

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-242600

(43)Date of publication of application : 07.09.1999

(51)Int.Cl.

G06F 9/46

(21)Application number : 09-248812

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 12.09.1997

(72)Inventor :
TAKEDA SATORU
TSUCHIDA TAIJI
NAGANUMA KAZUTOMO
YOKOSATO JUNICHI
SUZUKI YASUHIRO

(30)Priority

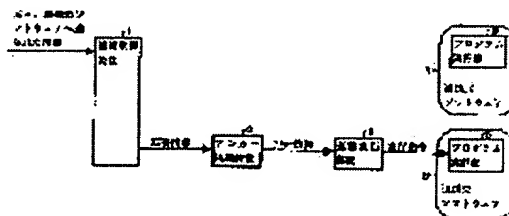
Priority number : 10520261 Priority date : 29.10.1996 Priority country : JP

(54) SOFTWARE COOPERATION SYSTEM FOR INFORMATION PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To extend the function of object software that has no or deficient extending function.

SOLUTION: An information processor having an operating system where plural types of software are operable is provided with an information acquiring device 1 which acquires the internal information on the information processor, an anchor identifying device 2 which decides the presence or absence of the anchor information that specifies the cooperating condition with which the cooperating software recognizes the cooperated software based on the information acquired by the device 1, and a cooperation executing device 3 which executes the cooperated software which performs a cooperating operation in response to the anchor information that is identified by the device 2. Thus, it's possible to obtain a software cooperation system of the information processor where both cooperating and cooperated software are operable by performing anchor processing outside the cooperation software without changing this software nor its file of forms.



LEGAL STATUS

[Date of request for examination] 12.09.1997

[Date of sending the examiner's decision of rejection] 18.07.2000

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-242600

(43) 公開日 平成11年(1999) 9月7日

(51) Int.Cl.⁸

G 0 6 F 9/46

識別記号

3 4 0

F I

G 0 6 F 9/46

3 4 0 B

審査請求 有 請求項の数18 O L (全 27 頁)

(21) 出願番号 特願平9-248812

(22) 出願日 平成9年(1997) 9月12日

(31) 優先権主張番号 特願平10-520261

(32) 優先日 平8(1996)10月29日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 武田 哲

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 土田 泰治

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 永沼 和智

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74) 代理人 弁理士 曾我 道照 (外6名)

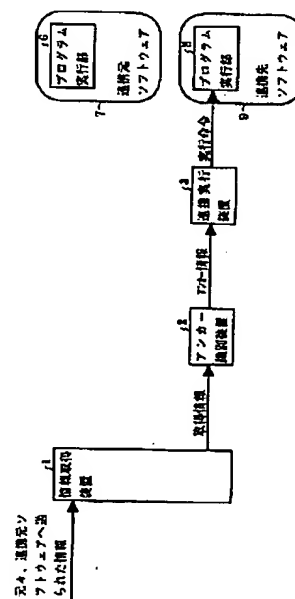
最終頁に続く

(54) 【発明の名称】 情報処理装置のソフトウェア連携システム

(57) 【要約】

【課題】 拡張機能のないもしくは拡張機能が不十分な対象ソフトウェアに対し機能拡張を行うことができる情報処理装置のソフトウェア連携システムを得る。

【解決手段】 複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、情報処理装置内の情報を取得する情報取得装置1と、上記情報取得装置1が取得した情報に基づいて連携元ソフトウェアが連携先ソフトウェアを認識するための連携条件を特定するアンカー情報の有無を判断するアンカー識別装置2と、アンカー識別装置により識別されたアンカー情報に対応して連携動作を行う連携先ソフトウェアを実行する連携実行装置3とを備えることにより、連携ソフトウェアやその形式のファイルを変更することなくアンカー処理を連携ソフトウェアの外部で行うことにより、ソフトウェアが連携元及び連携先の双方になることが可能な情報処理装置のソフトウェア連携システムを得る。



【特許請求の範囲】

【請求項1】 複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、情報処理装置内の情報を取得する情報取得手段と、上記情報取得手段が取得した情報に基づいて対象ソフトウェアの動作または対象ソフトウェアへの入力を監視して対象ソフトウェアに機能追加を行う機能追加手段とを備えた情報処理装置のソフトウェア連携システム。

【請求項2】 上記機能追加手段は、上記情報取得手段が取得した情報に基づいて対象ソフトウェアとしての連携元ソフトウェアが連携先ソフトウェアを認識するための連携条件を特定するアンカー情報の有無を判断するアンカー識別手段と、上記アンカー識別手段により識別されたアンカー情報に対応して連携動作を行う連携先ソフトウェアを実行する連携実行手段とを備えたことを特徴とする請求項1記載の情報処理装置のソフトウェア連携システム。

【請求項3】 上記情報取得手段は、上記情報処理装置のオペレーティングシステムが出す情報を連携元ソフトウェアに代わり受信すべく上記オペレーションシステムに対し情報取得登録を行う情報取得登録部と、上記オペレーティングシステムからの情報を受ける情報受信部とを備えたことを特徴とする請求項2記載の情報処理装置のソフトウェア連携システム。

【請求項4】 上記情報取得手段は、連携元ソフトウェアが画面表示するウィンドウを覆う透明ウィンドウ部と、上記透明ウィンドウ部に対し入力される情報を取得するウィンドウ情報取得部とを備えたことを特徴とする請求項2記載の情報処理装置のソフトウェア連携システム。

【請求項5】 上記情報取得手段は、取得する情報を登録蓄積してなる取得情報表記憶部と、上記取得情報表記憶部に登録された情報を取得する情報取得部とを備えたことを特徴とする請求項2記載の情報処理装置のソフトウェア連携システム。

【請求項6】 上記アンカー識別手段は、アンカー情報を登録蓄積してなるアンカー情報表記憶部と、上記情報取得手段が取得した情報と上記アンカー情報表記憶部にある情報とを照合するアンカー照合部とを備えたことを特徴とする請求項2記載の情報処理装置のソフトウェア連携システム。

【請求項7】 上記アンカー識別手段は、上記情報取得手段が取得した情報が条件を満たすアンカー情報がない場合に、情報を連携元ソフトウェアのプログラムへ送り出す情報送出部をさらに備えたことを特徴とする請求項6記載の情報処理装置のソフトウェア連携システム。

【請求項8】 上記アンカー識別手段は、上記アンカー情報表記憶部にアンカー情報を登録するアンカー登録部をさらに備えたことを特徴とする請求項6記載の情報処理装置のソフトウェア連携システム。

【請求項9】 上記アンカー識別手段は、上記情報取得手段が取得した情報を蓄積する取得情報蓄積手段をさらに備え、上記アンカー照合部は、上記取得情報蓄積手段に蓄積された複数の情報の組合せによるアンカー照合を可能にすることを特徴とする請求項6記載の情報処理装置のソフトウェア連携システム。

【請求項10】 上記アンカー識別手段は、上記アンカー照合部で条件を満たすと判断された全てのアンカー情報を蓄積して上記連携実行手段に出力する連携アンカー蓄積部をさらに備えたことを特徴とする請求項6記載の情報処理装置のソフトウェア連携システム。

【請求項11】 上記連携実行手段は、ソフトウェア連携情報を登録して蓄積する連携情報表記憶部と、上記アンカー識別手段から得られたアンカー情報に対応するソフトウェア連携情報を上記連携情報表記憶部から取得する連携情報取得部と、上記連携情報取得部が取得したソフトウェア連携情報から連携先ソフトウェアに実行命令を出す連携情報実行部とを備えたことを特徴とする請求項2記載の情報処理装置のソフトウェア連携システム。

【請求項12】 上記アンカー識別手段により識別または登録蓄積されたアンカー情報を画面に表示するアンカー表示手段をさらに備えたことを特徴とする請求項2記載の情報処理装置のソフトウェア連携システム。

【請求項13】 上記機能追加手段は、上記情報取得手段が取得した情報に基づいて対象ソフトウェアの動作が表示処理を行うトリガの場合に表示情報を出力する表示情報処理手段と、この表示情報処理手段からの表示情報を対象ソフトウェア上に表示する表示手段とを備えたことを特徴とする請求項1記載の情報処理装置のソフトウェア連携システム。

【請求項14】 操作情報を入力するための操作部と、操作命令情報を蓄積してなる操作命令情報記憶手段と、上記操作部から入力された操作情報に対応して実行すべき命令情報を上記操作命令情報記憶手段から検索し、検索された命令を実行する操作命令処理手段とをさらに備え、上記操作命令処理手段は、命令情報の中で対象ソフトウェア上に表示を行う際に、上記表示情報処理手段に表示処理を依頼することを特徴とする請求項13記載の情報処理装置のソフトウェア連携システム。

【請求項15】 上記操作命令処理手段により検索された命令情報が対象ソフトウェアに対する命令である場合に、当該操作命令処理手段から出力される命令情報の入力に基づいて対象ソフトウェアへの命令を送信する対象ソフトウェア命令手段をさらに備えたことを特徴とする請求項14記載の情報処理装置のソフトウェア連携システム。

【請求項16】 対象ソフトウェア使用時に高い頻度で行う処理の手順情報を蓄積してなる処理手順情報記憶手段と、指定された処理手順を上記処理手順情報記憶手段から検索する処理手順検索手段とをさらに備え、上記操

作命令処理手段は、上記操作部から入力された操作情報に対応する命令情報を検索し、操作命令情報が処理手順情報記憶手段に蓄積されている処理を行う命令であった場合に、上記処理手順検索手段に処理手順を指定して検索させることを特徴とする請求項15記載の情報処理装置のソフトウェア連携システム。

【請求項17】 操作命令情報記憶手段に蓄積する操作命令を追加する操作命令追加手段をさらに備えたことを特徴とする請求項14記載の情報処理装置のソフトウェア連携システム。

【請求項18】 複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、連携元のソフトウェアのファイルまたはウィンドウが表示する内容をアンカー識別できるソフトウェア形式に変換するファイル変換手段と、アンカー識別できるソフトウェアが識別したアンカー情報に対応して連携先ソフトウェアを実行する連携実行手段とを備えた情報処理装置のソフトウェア連携システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、複数のソフトウェアが同時に動作可能なオペレーティングシステムを持つ情報処理装置に係り、特に、動作中のソフトウェアの情報を取得して、他のソフトウェアを連携して動作させることが可能な情報処理装置のソフトウェア連携システムに関するものである。

【0002】

【従来の技術】図56はマイクロソフト社のソフトウェア連携技術であるObject Linking and Embedding（以下、OLEと称す）を用いたハイパーメディアタイプのソフトウェア連携システムの画面例を示す図である。ここで、ハイパーメディアとは、連携するための連携動作を行う連携先のソフトウェアを認識する情報を連携元にアンカー情報として組み込むことにより、連携を実現することである。すなわち、図56に示す画面例のように、連携元であるソフトウェアA112は、自分の形式のファイルAA113を開いたウィンドウ上にソフトウェアB114形式のファイルBB115の一部を張り付け表示し、ソフトウェアA112上のファイルBB115の部分にマウス111によるイベントがあった場合、連携先としてファイルBB115の元々のソフトウェアであるソフトウェアB114を起動しファイルBB115を開き、ウィンドウ表示を行う。

【0003】上述したように、OLEを用いた従来のソフトウェア連携システムは、連携元ソフトウェアA112形式のファイルAA113に予め連携先ソフトウェアB114形式のファイルBB115を張り付ける操作が必要で、張り付け内容をアンカー情報としてソフトウェア連携を行う。張り付け内容の情報や連携先ソフトウェ

アといった情報は、連携元ソフトウェアA112形式のファイルAA113に保持される。

【0004】

【発明が解決しようとする課題】しかしながら、ソフトウェアを連携するために必要であるアンカーを処理する機能はソフトウェア内部にしかなく、アンカー処理はソフトウェア毎別々に組み込まれ、実行されていた。したがって、従って、OLEの機能を持たないソフトウェアは、アンカー情報を保持することができないため、連携元ソフトウェアになることができなかった。

【0005】また、マイクロソフト社Wordのマクロ言語のように追加機能が記述可能なアプリケーションや、上述したOLE対応により、開放している機能を外側から使用できるアプリケーションについては機能の拡張を行うことが可能であるが、そのようなインターフェースが用意されていないもしくは拡張したい機能を実現するには不十分なインターフェースしか有しないアプリケーションに対して機能の追加を行うことはできなかった。

【0006】この発明は、上記のような問題点を解消するためになされたもので、拡張機能のないもしくは拡張機能が不十分な対象ソフトウェアに対して機能拡張を行うことができる情報処理装置のソフトウェア連携システムを得ることを目的とする。具体的には、連携ソフトウェアやその形式のファイルを変更することなく、アンカー処理を連携ソフトウェアの外部で行うことにより、ソフトウェアが連携元及び連携先の双方になることが可能な情報処理装置のソフトウェア連携システムを得ることを目的とする。また、対象ソフトウェアを監視し、対象ソフトウェアの動作が表示処理を行う際に表示情報を対象ソフトウェア上に表示することで、拡張機能のないもしくは拡張機能が不十分なソフトウェアに対して機能拡張を行うことができる情報処理装置のソフトウェア連携システムを得ることを目的とする。

【0007】

【課題を解決するための手段】この発明に係る情報処理装置のソフトウェア連携システムは、複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、情報処理装置内の情報を取得する情報取得手段と、上記情報取得手段が取得した情報に基づいて対象ソフトウェアの動作または対象ソフトウェアへの入力を監視して対象ソフトウェアに機能を追加を行う機能追加手段とを備えたものである。

【0008】また、上記機能追加手段は、上記情報取得手段が取得した情報に基づいて対象ソフトウェアとしての連携元ソフトウェアが連携先ソフトウェアを認識するための連携条件を特定するアンカー情報の有無を判断するアンカー識別手段と、上記アンカー識別手段により識別されたアンカー情報に対応して連携動作を行う連携先ソフトウェアを実行する連携実行手段とを備えたことを特徴とするものである。

10

20

30

40

50

【0009】また、上記情報取得手段は、上記情報処理装置のオペレーティングシステムが出す情報を連携元ソフトウェアに代わり受信すべく上記オペレーションシステムに対し情報取得登録を行う情報取得登録部と、上記オペレーティングシステムからの情報を受ける情報受信部とを備えたことを特徴とするものである。

【0010】また、上記情報取得手段は、連携元ソフトウェアが画面表示するウィンドウを覆う透明ウィンドウ部と、上記透明ウィンドウ部に対し入力される情報を取得するウィンドウ情報取得部とを備えたことを特徴とするものである。

【0011】また、上記情報取得手段は、取得する情報を登録蓄積してなる取得情報表記憶部と、上記取得情報表記憶部に登録された情報を取得する情報取得部とを備えたことを特徴とするものである。

【0012】また、上記アンカー識別手段は、アンカー情報を登録蓄積してなるアンカー情報表記憶部と、上記情報取得手段が取得した情報と上記アンカー情報表記憶部にある情報とを照合するアンカー照合部とを備えたことを特徴とするものである。

【0013】また、上記アンカー識別手段は、上記情報取得手段が取得した情報が条件を満たすアンカー情報がない場合に、情報を連携元ソフトウェアのプログラムへ送り出す情報送出部をさらに備えたことを特徴とするものである。

【0014】また、上記アンカー識別手段は、上記アンカー情報表記憶部にアンカー情報を登録するアンカー登録部をさらに備えたことを特徴とするものである。

【0015】また、上記アンカー識別手段は、上記情報取得手段が取得した情報を蓄積する取得情報蓄積手段をさらに備え、上記アンカー照合部は、上記取得情報蓄積手段に蓄積された複数の情報の組合せによるアンカー照合を可能にすることを特徴とするものである。

【0016】また、上記アンカー識別手段は、上記アンカー照合部で条件を満たすと判断された全てのアンカー情報を蓄積して上記連携実行手段に出力する連携アンカー蓄積部をさらに備えたことを特徴とするものである。

【0017】また、上記連携実行手段は、ソフトウェア連携情報を登録して蓄積する連携情報表記憶部と、上記アンカー識別手段から得られたアンカー情報に対応するソフトウェア連携情報を上記連携情報表記憶部から取得する連携情報取得部と、上記連携情報取得部が取得したソフトウェア連携情報から連携先ソフトウェアに実行命令を出す連携情報実行部とを備えたことを特徴とするものである。

【0018】また、上記アンカー識別手段により識別または登録蓄積されたアンカー情報を画面に表示するアンカー表示手段をさらに備えたことを特徴とするものである。

【0019】また、上記機能追加手段は、上記情報取得

手段が取得した情報に基づいて対象ソフトウェアの動作が表示処理を行うトリガの場合に表示情報を出力する表示情報処理手段と、この表示情報処理手段からの表示情報を対象ソフトウェア上に表示する表示手段とを備えたことを特徴とするものである。

【0020】また、操作情報を入力するための操作部と、操作命令情報を蓄積してなる操作命令情報記憶手段と、上記操作部から入力された操作情報に対応して実行すべき命令情報を上記操作命令情報記憶手段から検索し、検索された命令を実行する操作命令処理手段とをさらに備え、上記操作命令処理手段は、命令情報の中で対象ソフトウェア上に表示を行う際に、上記表示情報処理手段に表示処理を依頼することを特徴とするものである。

【0021】また、上記操作命令処理手段により検索された命令情報が対象ソフトウェアに対する命令である場合に、当該操作命令処理手段から出力される命令情報の入力に基づいて対象ソフトウェアへの命令を送信する対象ソフトウェア命令手段をさらに備えたことを特徴とするものである。

【0022】また、対象ソフトウェア使用時に高い頻度で行う処理の手順情報を蓄積してなる処理手順情報記憶手段と、指定された処理手順を上記処理手順情報記憶手段から検索する処理手順検索手段とをさらに備え、上記操作命令処理手段は、上記操作部から入力された操作情報に対応する命令情報を検索し、操作命令情報が処理手順情報記憶手段に蓄積されている処理を行う命令であった場合に、上記処理手順検索手段に処理手順を指定して検索させることを特徴とするものである。

【0023】また、操作命令情報記憶手段に蓄積する操作命令を追加する操作命令追加手段をさらに備えたことを特徴とするものである。

【0024】さらに、他の発明に係る情報処理装置のソフトウェア連携システムは、複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、連携元のソフトウェアのファイルまたはウィンドウが表示する内容をアンカー識別できるソフトウェア形式に変換するファイル変換手段と、アンカー識別できるソフトウェアが識別したアンカー情報に対応して連携先ソフトウェアを実行する連携実行手段とを備えたものである。

【0025】

【発明の実施の形態】実施の形態1。以下、この発明の実施の形態1について説明する。図1は実施の形態1に係るソフトウェア連携システムの構成図である。図1に示されるソフトウェア連携システムは、対象ソフトウェアとしての連携元ソフトウェア7と連携先ソフトウェア9とが同時に動作可能なオペレーションシステムを持つ情報処理装置において、動作中の連携元ソフトウェア7の情報を取得して、連携先ソフトウェア9を連携動作さ

せることのできるもので、情報処理装置内の情報、つまり、元々、連携元ソフトウェア7へ送られた情報を取得する情報取得装置1と、上記情報取得装置1が取得した情報に基づいて連携元ソフトウェア7が連携先ソフトウェア9を認識するための連携条件を特定するアンカー情報の有無を判断するアンカー識別装置2と、上記アンカー識別装置2により識別されたアンカー情報に対応して連携動作を行う連携先ソフトウェア9を実行するプログラム実行部8に実行命令を出力する連携実行装置3とを備えており、上記アンカー識別装置2及び上記連携実行装置3は、連携元ソフトウェア7に機能追加を行う機能追加手段をなす。なお、図1において、6は連携元ソフトウェア7を実行するプログラム実行部を示す。

【0026】図2は、ユーザが入力手段を用いて情報を入力することにより、ソフトウェアを連携動作させる画面例を示すもので、ユーザがソフトウェアA102に対してマウス101により入力を行ったときに、ソフトウェアB103のファイル104をウィンドウ表示する画面例であり、このとき、ソフトウェア連携システムは、図3のような動作フローとなる。なお、入力手段としては、マウス101の他に、例えば、キーボード、ペン、トラックボール等がある。

【0027】次に、本実施の形態に係るソフトウェア連携システムの動作を図3に示すフローチャートを用いて説明する。ユーザからの入力があった場合、まず、ステップS1で、情報取得装置1が入力情報を取得する。この入力情報は、元々はユーザが連携元ソフトウェア7に対し入力した情報であるものの、本ソフトウェア連携システムでは、情報取得装置1が連携元ソフトウェア7に代わって取得することを可能にする。入力情報は、情報処理装置1の入力装置から入力される情報であり、特に限定しない。例えば、マウスイベントの種類、マウスの入力座標、イベントの対象ソフトウェア名等がある。

【0028】次に、ステップS2で、アンカー識別装置2が入力情報を条件に含むアンカー情報の有無を識別する。アンカー情報とは、連携元ソフトウェア7が連携先ソフトウェア9を認識するための情報である。アンカー情報となる情報の種類は、上記入力情報の他にソフトウェア固有の情報、例えばオープン中のファイル名、ページ番号等がある。アンカー情報があった場合は、ステップS3で、連携実行装置3がアンカー情報に対応する連携情報の処理を命令する。連携情報となる情報の種類は、例えば連携先ソフトウェア名、オープンさせるファイル名等がある。

【0029】上述した図2の画面例では、まず、ソフトウェアA102に対するマウス101によるマウスイベントを、ソフトウェアA102に代わり情報取得装置1が取得し、次に、アンカー識別装置2がマウスイベントに対するアンカー情報の有無を確認する。最後に、連携実行装置3がアンカー情報に対応する、「ソフトウェア

B103を起動してファイル104を開く」という連携情報を取得し、この連携情報の処理を命令する。

【0030】ここで、上記情報取得装置1の内部構成及び動作フローを図4及び図5を用いて説明する。図4に示すように、情報取得装置1の内部構成としては、情報処理装置のオペレーティングシステム23が出す情報を連携元ソフトウェア7に代わり受信すべく上記オペレーティングシステム23に対し情報取得登録を行う情報取得登録部21と、上記オペレーティングシステム23からの情報を受ける情報受信部22とを備えている。

【0031】この情報取得装置1は、図5に示すように、本ソフトウェア連携システムが起動すると、まず、ステップS5で、情報取得登録部21からオペレーティングシステムへ、オペレーティングシステム23が持つ情報の送り先を本ソフトウェア連携システムとする登録を行う。オペレーティングシステム23は特に限定しないが、例えばUNIXやWindowsなどである。登録方法は、例えば、MS-Windowsのフック機能等がある。次に、情報のユーザからの入力情報があると、ステップS6で、情報受信部22がオペレーティングシステム23から情報を受け取り、ステップS7で取得情報をアンカー識別装置2へ送出する。

【0032】また、上記アンカー識別装置2の内部構成及び動作フローを図6及び図7を用いて説明する。図6に示すように、アンカー識別装置2の内部構成としては、アンカー情報を登録蓄積してなるアンカー情報表記憶部32と、情報取得装置1が取得した情報と上記アンカー情報表記憶部32にある情報とを照合するアンカー照合部31とを備えている。

【0033】このアンカー識別装置2は、図7に示すように、まず、ステップS10で、アンカー照合部31が情報取得装置1が取得した情報からアンカー情報となり得る情報を抽出する。次に、ステップS11で、アンカー情報表記憶部32に登録されたアンカー情報を取り出し、ステップS12で、取得情報がアンカー情報の条件を満たすかどうか確認する。取得情報がアンカー情報の条件を満たしている場合は、ステップS13で、アンカー情報に対応するアンカーIDを取得し、ステップS14で、アンカーIDを連携実行装置3へ送出する。

【0034】このとき、アンカー情報の種類をソフトウェア名とした場合のアンカー情報表記憶部32に記憶されているアンカー情報表としては、図8のように、アンカーIDに対応してソフトウェア名が書き込まれている。なお、このアンカー情報表は、ユーザが自由に定義変更可能なように、例えば、外部ファイルとして登録、蓄積することも可能である。

【0035】また、上記連携実行装置3の内部構成及び動作フローを図9及び図10を用いて説明する。図9に示すように、連携実行装置3の内部構成としては、ソフトウェア連携情報を登録して蓄積する連携情報表記憶部

43と、アンカー識別装置2から得られたアンカー情報（アンカーID）に対応するソフトウェア連携情報を上記連携情報表記憶部43から取得する連携情報取得部41と、上記連携情報取得部41が取得したソフトウェア連携情報から連携先ソフトウェアに実行命令を出す連携情報実行部42とを備えている。

【0036】この連携実行装置3は、図10に示すように、まず、ステップS16で、連携情報取得部41がアンカー情報識別装置2から送出されたアンカーIDに対応する連携情報を連携情報表記憶部43から取得する。次に、ステップS17で、連携情報実行部42が連携先ソフトウェア9のプログラム実行部8に連携情報の処理を実行させる命令を出す。

【0037】このときの連携情報表記憶部43に記憶される連携情報となる情報の種類は、図11に示すように、例えばアンカーIDに対応する連携先ソフトウェア名、オープンさせるファイル名、連携先ソフトウェアのウィンドウ状態がある。この中で、連携先ソフトウェア名等の連携先ソフトウェアを特定するための情報は必須情報で、その他の情報は設定されている場合のみ処理される。なお、この連携情報表記憶部43は、ユーザが自由に定義変更可能なように、例えば、外部ファイルとして登録、蓄積することも可能である。

【0038】従って、実施の形態1によれば、情報取得装置1で、オペレーティングシステム23から連携元ソフトウェア7への情報を代わりに取得し、アンカー識別装置2で、連携元ソフトウェア7の外部でアンカー情報を識別することにより、連携元ソフトウェア7自体に変更を加えることなくソフトウェア連携を実現することができる。また、アンカー識別装置2で、ソフトウェア名をアンカー情報の種類とすることにより、連携元ソフトウェア7を特定したソフトウェア連携が実現できる。これより、情報処理装置内で動作する全てのソフトウェアに対し個別に、ソフトウェア自体の変更なしに外部から連携処理を実行させることができるようになる。

【0039】実施の形態2。上述した実施の形態1では、情報取得装置1がオペレーティングシステム23から情報を受信する場合について説明したが、連携元ソフトウェア7が画面表示するウィンドウ上を覆う透明ウィンドウを表示し、連携元ソフトウェア7に対するウィンドウイベントを、連携元ソフトウェア7の代わりに透明ウィンドウから取得することも可能である。透明ウィンドウが連携元ソフトウェア7の表示画面を覆う際の覆う方は特に限定しない。

【0040】実施の形態2に係る情報取得装置1の内部構成及び動作フローを図12及び図13を用いて説明する。図12に示すように、情報取得装置1の内部構成図としては、連携元ソフトウェア7が画面表示するウィンドウを覆う透明ウィンドウ部26と、透明ウィンドウ部26に対し入力される情報を取得するウィンドウ情報取得

部27とを備えている。

【0041】この情報取得装置1は、図13に示すように、まず、ステップS20で、透明ウィンドウ部26が連携元ソフトウェア7のウィンドウ上に透明ウィンドウを表示する。ステップS21で、ウィンドウ情報取得部27が透明ウィンドウ部26に対するユーザから入力されたウィンドウイベントを取得し、ステップS22で取得した情報をアンカー識別装置2へ送り出す。

【0042】従って、実施の形態2によれば、情報取得装置1で、連携元ソフトウェア7への情報を取得するため透明ウィンドウ部26を用いることにより、オペレーティングシステム23から連携元ソフトウェア7への情報を代わりに取得できない場合でも、ソフトウェア連携が実現できる。

【0043】実施の形態3。上述した実施の形態1では、情報取得装置1がオペレーティングシステム23から情報を受信する場合について説明したが、予め登録された情報を連携元ソフトウェア7から取得することも可能である。

【0044】実施の形態3に係る情報取得装置1の内部構成及び動作フローを図14及び図15を用いて説明する。図14に示すように、情報取得装置1の内部構成としては、取得する情報を登録蓄積してなる取得情報表記憶部20と、取得情報表記憶部20に登録された情報を取得する情報取得部19とを備えている。

【0045】この情報取得装置1は、図15に示すように、まず、ステップS26で、情報取得部19が取得情報表記憶部20に登録されたソフトウェアが実行中か否かを確認する。実行中の場合、ステップS27で、連携元ソフトウェア7から取得情報表記憶部20に取得情報として登録され情報を取得し、ステップS28で、アンカー識別装置2へ情報を送り出す。

【0046】このとき、連携元ソフトウェア7の表示するウィンドウのオブジェクトから情報を取得する場合の取得情報表記憶部20内の取得情報表の内容は図16のようになる。この中で、連携元ソフトウェア名等の連携先ソフトウェアを特定するための情報は必須情報で、その他の情報は設定されている場合のみ処理される。なお、この取得情報表記憶部20は、ユーザが自由に定義、変更可能なように、例えば、外部ファイルとして登録蓄積することも可能である。

【0047】従って、実施の形態3によれば、また、情報取得装置1で、予め取得情報を設定することにより、オペレーティングシステムからは取得できない情報、例えば、連携元ソフトウェア7が外部に公開している情報を連携元ソフトウェア7から受け取ったり、連携元ソフトウェア7の表示内容を取得した情報を元に、ソフトウェア連携が実現できる。

【0048】実施の形態4。また、上述した実施の形態3では、連携元ソフトウェアの表示するウィンドウのオ

10

20

30

40

50

プロジェクトから情報を取得する場合1について説明したが、連携元ソフトウェアが外部から参照可能な情報を受け取ることも可能である。この場合、図14に示す情報取得装置1の構成において、実施の形態4に係る取得情報記憶部20は、図17に示す如く、連携元ソフトウェア名に対応する変数名及び取得内容を登録蓄積する。

【0049】実施の形態5。また、上述した実施の形態3では、連携元ソフトウェアの表示するウィンドウのオブジェクトから情報を取得する場合について説明したが、情報処理装置内の特定のアドレスから情報を取得することも可能である。この場合、図14に示す情報取得装置1の構成において、実施の形態4に係る取得情報記憶部20は、図18に示す如く、取得アドレスに対応する取得内容を登録蓄積する。

【0050】実施の形態6。また、上述した実施の形態3では、連携元ソフトウェアの表示するウィンドウのオブジェクトから情報を取得する場合について説明したが、連携元ソフトウェアが実行中出力する情報がある場合、その情報からアンカー情報を取得することも可能である。この場合、図14に示す情報取得装置1の構成において、実施の形態4に係る取得情報記憶部20は、図19に示す如く、ファイル名に対応する取得内容を登録蓄積する。

【0051】実施の形態7。次に、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、イベントをアンカー情報の種類とすることも可能である。イベントは、例えば、マウスボタンダウン、マウス移動、キー入力、ウィンドウのアクティブ化等がある。また、イベントの発生元は、例えば、ユーザ入力、ソフトウェア処理、遠隔マシンからのネットワーク経由等がある。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図20に示す如く、アンカーIDに対応するイベントを登録蓄積する。

【0052】実施の形態8。また、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、マウス、トラックボール等の入力手段により入力された座標をアンカー情報の種類とすることも可能である。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図21に示す如く、アンカーIDに対応する座標を登録蓄積する。

【0053】実施の形態9。さらに、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、キーボード、マウス、ペン等の入力手段により入力されたキー情報、マウスやペンのボタン情報、ダブルクリック、シングルクリック等の情報をアンカー情報の種

類とすることも可能である。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図22に示す如く、アンカーIDに対応するキー情報を登録蓄積する。

【0054】従って、実施の形態7ないし9によれば、アンカー識別装置2で、イベント、座標、キー情報といったソフトウェア以外の情報をアンカー情報の種類とすることにより、ソフトウェアに依存しない、情報処理装置内で統一された操作によるソフトウェア連携が実現できる。

【0055】実施の形態10。次に、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、ウィンドウの状態情報をアンカー情報の種類とすることも可能である。ウィンドウの状態情報は、例えば、アイコン化、最大表示化、アクティブ化、ページ移動、カーソル移動等である。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図23に示す如く、アンカーIDに対応するウィンドウ状態を登録蓄積する。

【0056】従って、実施の形態10によれば、アンカー識別装置2で、ウィンドウ状態情報をアンカー情報の種類とすることにより、ユーザ入力がない場合であっても、ウィンドウの状態遷移を元にした自動的なソフトウェア連携が実現できる。

【0057】実施の形態11。次に、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、ユーザ入力を受けるウィンドウのオブジェクト名をアンカー情報の種類とすることも可能である。オブジェクト名は、ウィンドウ内で認識可能な情報全てであり、例えば、ボタン、ラベル等がある。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図24に示す如く、アンカーIDに対応するオブジェクト名を登録蓄積する。

【0058】実施の形態12。また、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、ウィンドウが表示するオープン中のファイル名をアンカー情報の種類とすることも可能である。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図25に示す如く、アンカーIDに対応するオープンファイル名を登録蓄積する。

【0059】実施の形態13。さらに、上述した実施の形態1では、アンカー識別装置2におけるアンカー情報の種類としてソフトウェア名を使う場合について説明したが、ソフトウェア毎にもつ情報をアンカー情報の種類とすることも可能である。ソフトウェア毎にもつ情報

10

20

30

40

50

は、例えば、ウィンドウが表示するページ番号、フレーム番号等がある。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図26に示す如く、アンカーIDに対応するページ番号を登録蓄積する。

【0060】従って、実施の形態11ないし13によれば、アンカー識別装置2で、オブジェクト名、ファイル名、ページ番号といったソフトウェア毎に持つ情報をアンカー情報の種類とすることにより、よりソフトウェアの動作に合わせたソフトウェア連携が実現できる。

【0061】実施の形態14、さらに、上述したアンカー情報の種類の一部または全部を合わせたアンカー情報表記憶部32に登録蓄積することも可能である。この場合、図6に示すアンカー識別装置2内のアンカー情報表記憶部32が登録蓄積するアンカー情報表としては、図27に示す如く、アンカーIDに対応するソフトウェア名、イベント、座標、キー情報、ウィンドウ状態、オブジェクト名、オープンファイル名及びページ番号を登録蓄積する。

【0062】従って、実施の形態14によれば、アンカー識別装置2で、アンカー情報の種類の一部または全部を組み合わせたアンカー情報とすることにより、より多くのパターンのアンカー登録が可能となり、より細かなソフトウェア連携が実現できる。

【0063】実施の形態15、次に、図6に示す実施の形態1に係るアンカー識別装置2に、アンカー情報表32の条件に含まれない取得情報を連携元ソフトウェア7へ送出することができるようにするために、情報送出部を備えることも可能である。すなわち、実施の形態15に係るアンカー識別装置2の内部構成として、図28に示すように、アンカー照合部31及びアンカー情報表記憶部32の他に、情報取得装置1が取得した情報が条件を満たすアンカー情報がない場合に、情報を連携元ソフトウェア7のプログラムへ送り出す情報送出部33をさらに備える。

【0064】従って、実施の形態15によれば、さらに、アンカー識別装置2で、アンカー情報の条件に含まれない取得情報を連携元ソフトウェア7へ送出することにより、連携元ソフトウェアの処理も可能となる。このため、操作の向上が期待できる。

【0065】実施の形態16、また、図6に示す実施の形態1に係るアンカー識別装置2に、外部からいってもアンカー情報表記憶部32にアンカー情報を登録できるようにするため、アンカー登録部を備えることも可能である。すなわち、実施の形態16に係るアンカー識別装置2の内部構成として、図29に示すように、アンカー照合部31及びアンカー情報表記憶部32の他に、アンカー情報表記憶部32にアンカー情報を登録するアンカー登録部34をさらに備える。

【0066】従って、実施の形態16によれば、アンカ

ー識別装置2で、いつでも外部からアンカー登録できることにより、利用状況に合わせたソフトウェア連携を実現できるようになる。このため、操作の向上が期待できる。

【0067】実施の形態17、また、図6に示す実施の形態1に係るアンカー識別装置2に、複数の入力情報の組み合わせをアンカー情報とすることにより、取得情報を蓄積する取得情報蓄積部を備えることも可能である。すなわち、実施の形態17に係るアンカー識別装置2の内部構成として、図30に示すように、アンカー照合部31及びアンカー情報表記憶部32の他に、情報取得装置1が取得した情報を蓄積する取得情報蓄積部35をさらに備え、アンカー照合部31により、取得情報蓄積部35に蓄積された複数の情報の組合せによるアンカー照合を可能にする。

【0068】従って、実施の形態17によれば、アンカー識別装置2で、複数の情報の組み合わせをアンカー情報とすることにより、独自に設定した操作へのソフトウェア連携が可能となる。このため、通常、行わない操作へ連携を登録することができ、操作誤りを原因とする、ソフトウェア連携の誤動作が減少できると期待できる。

【0069】実施の形態18、また、図6に示す実施の形態1に係るアンカー識別装置2に、複数の連携を同時あるいは逐次に行うようにするために、取得情報が条件に含まれる複数のアンカー情報を蓄積する連携アンカー蓄積部を備えることも可能である。すなわち、実施の形態18に係るアンカー識別装置2の内部構成として、図31に示すように、アンカー照合部31及びアンカー情報表記憶部32の他に、アンカー照合部31で条件を満たすと判断された全てのアンカー情報（取得情報を条件に含むアンカー情報のアンカーID）を蓄積して連携実行装置3に出力する連携アンカー蓄積部36をさらに備える。

【0070】従って、実施の形態18によれば、アンカー識別装置2で複数のアンカー情報を蓄積することにより、複数の連携処理を実行することができる。

【0071】実施の形態19、また、上述した実施の形態1において、アンカー識別装置2により識別または登録蓄積されたアンカー情報を画面表示してユーザにアンカー情報を伝えるアンカー表示装置を備えることも可能である。すなわち、実施の形態19に係るソフトウェア連携システムとしては、図1に示す実施の形態1に対して、図32に示すように、アンカー情報を表示するアンカー表示装置4をさらに備える。アンカー表示装置4は、全てのアンカー情報を表示すること、一部のアンカー情報を表示することが可能である。また、アンカー識別装置2で識別されたアンカー情報のみを表示することが可能である。

【0072】従って、実施の形態19によれば、アンカー情報を画面表示することにより、ソフトウェア連携を

10

20

30

40

50

行う場所を視覚的にユーザへ伝えることができ、このため、操作の向上が期待できる。

【0073】実施の形態20. 次に、実施の形態20に係る他の発明による情報処理装置のソフトウェア連携システムの構成及び動作フローを図33及び図34を用いて説明する。この他の発明に係る情報処理装置のソフトウェア連携システムは、複数のソフトウェアが同時に動作可能なオペレーティングシステムを持つ情報処理装置であって、図33に示すように、連携元ソフトウェアのファイルまたはウィンドウが表示する内容をアンカー識別

【0074】上記ファイル変換装置10は、アンカー識別できないソフトウェアのウィンドウ表示内容11またはアンカー識別できないソフトウェア形式のファイル12をアンカー識別できるソフトウェア形式のファイル13に変換し、上記連携実行装置3は、アンカー識別できるソフトウェア15のプログラム実行部14が識別したアンカー情報に対応した連携先ソフトウェア9のプログラム実行部8に実行命令を出力する。

【0075】すなわち、この他の発明に係る情報処理装置のソフトウェア連携システムは、図34に示すように、まず、ステップS30で、ファイル変換装置10がアンカー識別できないソフトウェアのファイル12やアンカー識別できないソフトウェア形式のウィンドウ表示内容11からアンカー識別できるソフトウェア形式のファイル13に変換する。アンカー識別できるソフトウェア形式としては、例えば、ビットマップ形式、AVI形式等がある。次に、ステップS31で、アンカー識別できるソフトウェア15を起動してアンカー識別できるソフトウェア形式のファイル13をオープンし、アンカー識別できるソフトウェア15からアンカー情報が送られてきた場合、連携実行装置3がアンカー情報に対応する連携情報を取得し、この連携情報の処理を命令する。

【0076】従って、上記実施の形態20によれば、アンカー識別できないソフトウェア形式のファイル12やウィンドウ表示内容11をアンカー識別できるソフトウェア形式のファイル13に変換することにより、連携元ソフトウェア7自体に変更を加えることなくソフトウェア連携を実現できる。

【0077】実施の形態21. 次に、図35は実施の形態21に係るソフトウェア連携システムの構成図である。図35に示されるソフトウェア連携システムは、監視の対象となるソフトウェア（監視対象ソフトウェア）の動作を監視し、監視対象ソフトウェアの動作を表示情報処理手段201へ通知する情報取得手段200と、通知された動作が表示処理を行うトリガだった場合に表示

手段202へ表示情報を送信する表示情報処理手段201及び表示情報処理手段201から受けた表示情報を監視対象ソフトウェア上に表示する表示手段202でなる機能追加手段とを備えており、これにより、監視対象ソフトウェアに対して変更を加えることなく、表示機能を追加することが可能になっている。

【0078】前述したように、例えば Microsoft Windows 95オペレーティングシステムの場合、フック機能等により監視対象ソフトウェアの情報を取得することが可能である。また、透明なウィンドウ等で監視対象ソフトウェアの表示部上を覆うことで、監視対象ソフトウェア上に表示を行うことが可能である。以下、この2つの方法を例として、本実施の形態に対する説明を行う。

【0079】まず、情報取得手段200の内部構成及び動作フローを図36及び図37を用いて説明する。図36に示すように、情報取得手段200の内部構成としては、オペレーティングシステムに対し情報取得の登録を行う情報取得登録部210と、対象となるソフトウェアに対して送信されるメッセージ情報を受信する情報受信部211と、対象ソフトウェアを起動させる対象ソフトウェア起動部223とを備えている。

【0080】この情報取得手段200は、図37に示すように、本システムが起動すると、対象ソフトウェア起動部223で監視対象ソフトウェアを起動する（ステップS50）。監視対象ソフトウェアが起動されると、システムが動作しているオペレーティングシステムへ情報取得登録部210が情報取得の登録を行い（ステップS51）、対象となるソフトウェアに対して送信されるメッセージ情報を情報受信部211で受信する（ステップS52）。受信したメッセージは監視対象ソフトウェアのイベントを発生させるためのものであり、メッセージを得ることにより監視対象ソフトウェアに発生するイベント情報を監視することが可能となる。情報受信部211が受信した取得情報は表示情報処理手段201へ送信される（ステップS53）。

【0081】このように、情報取得手段200は、監視対象ソフトウェアへ送信される情報を受けると、表示情報処理手段201へ送信する。表示情報処理手段201は、情報取得手段200から送信されたメッセージを解析し、そのメッセージが表示を行うためのトリガである場合には、表示手段202へ表示情報を送信する。

【0082】次に、表示情報処理手段201の内部構成及び動作フローを図38及び図39を用いて説明する。図38に示すように、表示情報処理手段201の内部構成としては、情報取得手段200から受信した監視対象ソフトウェアに関する情報を解析する情報解析部212と、情報取得手段200から送信されてきた情報が対象ソフトウェアに対して表示を行うトリガになる情報である場合に表示情報を作成する表示情報作成部213と、情報解析部212からの入力に基づいて表示手段202

10

20

30

40

50

を制御するための表示手段制御部214とを備えている。

【0083】この表示情報処理手段201は、図39に示すように、情報取得手段200から受信した監視対象ソフトウェアに関する情報を情報解析部212で入力し（ステップS60）、この情報解析部212において、受信した情報に関する詳細を解析する。この情報には監視対象ソフトウェアで発生するイベントのメッセージに関する情報と同一の情報が含まれており、それを解析することで処理は表示情報を作成する場合と、表示手段202を制御する場合に別れる。

【0084】情報取得手段200から送信されてきた情報が対象ソフトウェアに対して表示を行うトリガになる情報であれば（ステップS60）、表示情報を作成する場合であり、表示情報作成部213が表示情報の作成を行う。表示情報作成部213は送信された情報に対して表示情報を作成し、表示手段202に表示命令を行う（ステップS62）。

【0085】情報取得手段200から送信されてきた情報が監視対象ソフトウェアの表示部の移動やリサイズに該当するメッセージであれば（ステップS63）、表示手段202を制御する場合であり、表示手段制御部214に表示手段制御を依頼する（ステップS64）。表示手段202として用いている透明ウィンドウによる監視対象ソフトウェアの表示ウィンドウの覆いかたは問わないが、例えば監視対象ソフトウェアの表示ウィンドウの上を表示ウィンドウと同一のサイズで覆っている場合には、表示手段制御部214は監視対象ソフトウェアの表示部が移動した場合に、監視対象ソフトウェアの表示部に対して、表示手段202として覆っている透明ウィンドウの位置情報を変更し、監視対象ソフトウェアの表示部に追従させる。

【0086】また、例えば表示手段制御部214は監視対象ソフトウェアの表示部がリサイズされた場合には、表示手段202として覆っている透明ウィンドウの大きさを監視対象ソフトウェアの表示部と同じ大きさに変更する。透明ウィンドウ上に直接送信される利用者の操作メッセージについては元々監視対象ソフトウェアに対する操作であるため、覆っているウィンドウに対して再送する。

【0087】表示手段202は表示情報処理手段201から受けた表示情報を透明ウィンドウ上に表示する。表示手段202が表示している表示情報以外は対象ソフトウェアの表示部が表示している情報のため、表示手段202が表示している表示情報は利用者からの見かけ上、対象ソフトウェアの表示部が表示しているように感じられる。

【0088】このように、情報取得手段が取得した情報に基づき監視対象の状態を監視し、表示手段により監視対象ソフトウェア上に表示を行うことで、拡張機能のな

いソフトウェアに対して機能拡張を行うことができる。

【0089】実施の形態22. 次に、図40は実施の形態22に係るソフトウェア連携システムの構成図を示し、図35に示す実施の形態21の構成に対して、操作情報を入力するための操作部203と、操作部203から入力された操作情報に対応して実行すべき命令情報を操作命令情報記憶手段205から検索し、検索された命令を実行する操作命令処理手段204と、操作命令情報を蓄積してなる操作命令情報記憶手段205とを追加したもので、表示情報処理手段201は、操作部203から入力される利用者の操作により操作命令処理手段204によって検索された操作命令情報に従って情報表示を行う。

【0090】ここで、操作部203は、表示手段202と同一の透明ウィンドウ内で実現し、操作するオブジェクトを表示手段202が表示する。操作部203が透明ウィンドウ上に実現されているため、表示上、利用者は透明ウィンドウが覆っている監視対象ソフトウェアの表示ウィンドウ上に操作部203があるように感じ、操作部203を操作する際に、利用者はその監視対象ソフトウェアの表示ウィンドウを操作しているように感じる。

【0091】上記操作部203は図41に示すフローチャートにしたがって動作する。すなわち、利用者が操作部203により操作を行うと（ステップS70）、操作部203は、その操作に対応する操作IDを図42に示す如く内蔵する操作／操作ID変換テーブル224を検索することにより得る（ステップS71）。操作部203で操作された情報は操作命令処理手段204へ通知される（ステップS72）。これにより、操作命令処理手段204は、操作部203から操作された操作IDを受けると、その操作IDに対応して実行すべき情報を操作命令情報記憶手段205から検索することになる。

【0092】また、上記操作命令処理手段204は図43に示すフローチャートにしたがって動作する。例えば図42に示すごとく操作部203に作成されている操作オブジェクトとしての「ボタン1」を押下したとする。操作命令処理手段204は、操作部203から「ボタン1」押下の操作IDとして「ID1」を受け取り（ステップS80）、その操作IDに該当する操作命令を、操作IDに対応する操作命令が登録されているデータベースである操作命令情報記憶手段205から検索する（ステップS81）。

上記操作命令情報記憶手段205は図44に示す如く操作ID毎にコマンドテーブルを持っており、操作IDから命令情報を検索する。操作命令処理手段204は、検索したコマンドテーブル中の命令を実行する（ステップS82）。その命令情報の中で監視対象ソフトウェア上に表示を行う場合には（ステップS83）、表示情報処理手段201に表示処理を依頼する（ステップS84）。これを操作命令が終了するまで続ける（ステップS85）。

【0093】従って、操作部203から入力された操作情報に対応して実行すべき命令情報を操作命令情報記憶手段205から検索し、検索された操作命令情報に従って表示情報処理手段201により情報表示を行うことができる。

【0094】実施の形態23. 次に、図45は実施の形態23に係るソフトウェア連携システムの構成図を示し、図40に示す実施の形態22の構成に対して、操作命令処理手段204により検索された命令情報が監視対象ソフトウェアに対する命令である場合に、当該操作命令処理手段204から出力される命令情報の入力に基づいて対象ソフトウェアへの命令を送信する対象ソフトウェア命令手段206を追加したものである。

【0095】上述のように、操作部203から入力された操作情報を用いて操作命令処理手段204は、操作命令情報記憶手段205からその操作に対する命令情報を検索する。検索された命令情報が現在の監視対象ソフトウェアに対する命令である場合には、対象ソフトウェア命令手段206により、監視対象ソフトウェアに対して命令を送信する。

【0096】例えば Windows95OSの場合には監視対象ソフトウェアの表示部のウィンドウの位置や大きさ等を外部から変更することができる。操作命令がこの様に標準的な方法により実現できる場合には監視対象ソフトウェアに対する制約はない。また、特に監視対象ソフトウェアが通信機能を持ち、外部からの命令を受信し、その命令を実行できるようなスクリプト機能を有している場合、監視対象ソフトウェアに対して命令を送信し、さらに細かい機能を、監視対象ソフトウェアに対して追加することが可能である。

【0097】上記操作命令処理手段204は図53に示すフローチャートにしたがって動作する。すなわち、操作命令処理手段204は、ユーザの操作により操作命令情報記憶手段205の検索を行い(ステップS90)、検索した命令情報を実行する(ステップS91)。検索した命令情報の内容が監視対象ソフトウェアに対する命令情報だった場合には(ステップS92)、対象ソフトウェア命令手段206に対して命令情報を送信する(ステップS93)。この過程を命令情報の全てを処理するまで繰り返す(ステップS94)。

【0098】また、上記対象ソフトウェア命令手段206の内部構成及び動作フローを図47及び図48を用いて説明する。図47に示すように、対象ソフトウェア命令手段206の内部構成としては、受信された命令情報が全てのソフトウェアに対して共通の命令である場合に対象ソフトウェアに対して全てのソフトウェア共通の方法で命令を送信する共通命令送信部218と、受信された命令情報が対象ソフトウェア個別の命令である場合に対象ソフトウェア毎に異なる方法で命令を送信する個別命令送信部219とを備えている。

【0099】この対象ソフトウェア命令手段206は、図48に示すように、命令情報を受信し(ステップS100)、その命令内容により、全てのソフトウェアに対して共通の命令である場合(ステップS101)には共通命令送信部218に命令を渡し(ステップS102)、対象とするソフトウェア個別の命令の場合には個別命令送信部219へ命令を渡し(ステップS103)。共通命令送信部218は、対象としているソフトウェアに対して全ソフトウェア共通の方法で命令を送信する。また、個別命令送信部219は、監視対象ソフトウェア毎に異なり、Windowsの場合にはDDEやOLEにより命令処理を行う。

【0100】これにより、監視対象ソフトウェアを操作している感覚で、監視対象ソフトウェア自体の状態を変化させるような機能を監視対象ソフトウェア自体の変更を行うことなく追加することができる。

【0101】実施の形態24. 次に、図49は実施の形態24に係るソフトウェア連携システムの構成図を示し、図45に示す実施の形態23の構成に対して、操作命令処理手段204から指定された処理手順を処理手順情報記憶手段208から検索する処理手順検索手段207と、対象ソフトウェア使用時に高い頻度で行う処理の手順情報を蓄積してなる処理手順情報記憶手段208とを追加したもので、操作命令処理手段204は、操作部203から入力された操作情報に対応する命令情報を検索し、操作命令情報が処理手順情報記憶手段208に蓄積されている処理を行う命令であった場合に、処理手順検索手段207に処理手順を指定して検索させる。

【0102】図49に示す実施の形態23に係るソフトウェア連携システムは、操作部203から入力された利用者の操作に対する操作ID検索時に、図42に示す操作/操作ID変換テーブルに対し、さらに手順情報が付加された操作部203に内蔵されている図50に示す操作/操作ID変換テーブルを検索し、操作IDと手順情報に基づいてその操作に対する実行が処理手順情報記憶手段208の実行であるかを検索する。操作命令情報が処理手順情報記憶手段208に蓄積されている処理を行う命令であった場合には、処理手順検索手段207により指定された処理手順を処理手順情報記憶手段208から検索し、処理手順の第1番目の動作命令を操作命令処理手段204から対象ソフトウェア命令手段206を通じて監視対象ソフトウェアに対して送信する。操作部203から次の処理に進む操作情報が来た場合には、操作命令処理手段204がさらに次の動作命令を送信する。

【0103】この実施の形態24に係るソフトウェア連携システムの動作を図51に示すフローチャートを参照して詳細に説明する。操作部203の手順操作オブジェクトにより操作された命令に対しては、操作命令処理手段204が該当する処理手順を処理手順検索手段207に検索依頼を行う(ステップS110)。処理手順情

報記憶手段208は、利用者が監視対象ソフトウェアを中心に行う一連の手順項目が登録されている。処理手順情報記憶手段208の一つの情報は複数の操作命令情報により構成されており、各操作命令情報が手順の1フェーズに対応している。処理手順検索手段207は、検索すべき手順情報が既に実行中かどうかを確認する(ステップS111)。未だ実行中でない場合には手順情報を処理手順情報記憶手段208から検索し、検索後、処理を行ったフェーズの次のフェーズにポインタを進め、ポインタ管理を行う(ステップS113)。既に実行中の処理手順情報に対しての検索要求が来た場合には、ポインタが示すフェーズの情報を返し、ポインタを1フェーズ進める(ステップS112)。処理手順検索手段207は、最終フェーズまで処理が行われた処理手順情報については開放する(ステップS116)。検索した1フェーズづつの手順情報は操作命令情報と同じため、操作命令情報を処理する方法と同様の方法で実行を行う(ステップS115)ことができる。

【0104】これにより、通常頻繁に実行を行う手順情報を保存しておき、拡張機能のないソフトウェアについてもそのソフトウェア自体を操作する感覚で、その手順情報を実行することができる。

【0105】実施の形態25。次に、図52は実施の形態25に係るソフトウェア連携システムの構成図を示し、図40に示す実施の形態22の構成に対して、操作命令情報記憶手段205に蓄積する操作命令を新たに追加するするための操作命令追加手段209を追加したものである。

【0106】操作命令追加手段209の内部構成及び動作フローを図53及び図54を用いて説明する。図53に示すように、操作命令追加手段209の内部構成としては、操作部203から追加される操作命令情報を入力する追加命令入力部220と、追加された操作命令情報を操作命令情報記憶手段205に登録するための追加命令登録部221と、操作部203への新たな操作IDの追加を行う操作部追加部222とを備えている。

【0107】この操作命令追加手段209は、図54に示すように、利用者により追加命令入力部220を利用して追加する命令情報が入力されると(ステップS120)、操作部追加部222を利用して操作部203に新たな操作IDを追加する(ステップS121)。これら操作IDと追加命令を対応づけし(ステップS122)、追加命令登録部221により操作命令情報記憶手段205に登録を行う(ステップS123)。

【0108】ここで、操作命令情報記憶手段205は、図55に示すように、参照テーブル215と、外部コマンドテーブル216と、内部コマンドテーブル217とに分割されている。上記参照テーブル215は、操作IDの命令が内部コマンドテーブル217に存在するか、外部コマンドテーブル216に存在するかを示すテ

ルであり、上記外部コマンドテーブル216は、操作命令を登録しておくプログラムの外部に存在するテーブルであり、さらに、上記内部コマンドテーブル217は、プログラム中に登録されている操作命令である。

【0109】もともとシステムが提供している機能については、参照テーブル215にInternalが登録されており、内容は内部コマンドテーブル217に格納されている。新たにユーザにより追加された機能については、参照テーブル215にExternalが登録されており、内容は外部コマンドテーブル216に登録されている。操作命令追加手段209により追加を行った操作命令情報記憶手段205の参照テーブル215に登録を行ったIDは参照先がExternalとなり、図44に示す外部コマンドテーブルに命令情報が追加される。ここで、利用者が登録する追加命令の形態は特に問わない。例えば一般的なマクロの形態をとることもできる。

【0110】このように、操作命令追加手段209により、利用者が新たに操作命令情報を追加し、またその操作命令情報のトリガを発生させる操作部203のオブジェクトを追加することができる。

【0111】

【発明の効果】以上のように、この発明に係る情報処理装置のソフトウェア連携システムによれば、複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、情報処理装置内の情報を取得する情報取得手段と、上記情報取得手段が取得した情報に基づいて対象ソフトウェアの動作または対象ソフトウェアへの入力を監視して対象ソフトウェアに機能を追加を行う機能追加手段とを備えたので、拡張機能のない対象ソフトウェアに対して機能拡張を行うことができる。

【0112】また、上記機能追加手段は、上記情報取得手段が取得した情報に基づいて対象ソフトウェアとしての連携元ソフトウェアが連携先ソフトウェアを認識するための連携条件を特定するアンカー情報の有無を判断するアンカー識別手段と、上記アンカー識別手段により識別されたアンカー情報に対応して連携動作を行う連携先ソフトウェアを実行する連携実行手段とを備えたので、情報取得手段が取得した情報に基づいて連携元ソフトウェアが連携先ソフトウェアを認識するための連携条件を特定するアンカー情報の有無をアンカー識別手段により判断し、連携実行手段によりそのアンカー情報に対応して連携動作を行う連携先ソフトウェアを実行することで、アンカーを処理する機能がなくとも、ソフトウェア連携における連携元となることを可能になり、連携ソフトウェアやその形式のファイルを変更することなく、アンカー処理を連携ソフトウェアの外部で行うことにより、ソフトウェアが連携元及び連携先の双方になることが可能となる。

【0113】また、上記情報取得手段は、上記情報処理装置のオペレーティングシステムが出す情報を連携元ソ

10

20

30

40

50

ソフトウェアに代わり受信すべく上記オペレーションシステムに対し情報取得登録を行う情報取得登録部と、上記オペレーティングシステムからの情報を受ける情報受信部とを備えたので、オペレーティングシステムから連携元ソフトウェアへの情報を代わりに取得し、アンカー識別装置で連携元ソフトウェアの外部でアンカー情報を識別することで、連携元ソフトウェア自体に変更を加えることなくソフトウェア連携を実現できる。

【0114】また、上記情報取得手段は、連携元ソフトウェアが画面表示するウィンドウを覆う透明ウィンドウ部と、上記透明ウィンドウ部に対し入力される情報を取得するウィンドウ情報取得部とを備えたので、オペレーティングシステムから連携元ソフトウェアへの情報を代わりに取得できない場合でもソフトウェア連携が実現できる。

【0115】また、上記情報取得手段は、取得する情報を登録蓄積してなる取得情報表記憶部と、上記取得情報表記憶部に登録された情報を取得する情報取得部とを備えたので、予め登録された情報を連携元ソフトウェアから取得することが可能になり、予め取得情報を設定することにより、オペレーティングシステムからは取得できない情報を元にソフトウェア連携が実現できる。

【0116】また、上記アンカー識別手段は、アンカー情報を登録蓄積してなるアンカー情報表記憶部と、上記情報取得手段が取得した情報と上記アンカー情報表記憶部にある情報とを照合するアンカー照合部とを備えたので、蓄積された複数の情報の組み合わせによるアンカー照合を可能にする。

【0117】また、上記アンカー識別手段は、上記情報取得手段が取得した情報が条件を満たすアンカー情報がない場合に、情報を連携元ソフトウェアのプログラムへ送り出す情報送出部をさらに備えたので、アンカー情報の条件に含まれない取得情報を連携元ソフトウェアへ送出することにより、連携元ソフトウェアの処理も可能となり、操作の向上が期待できる。

【0118】また、上記アンカー識別手段は、上記アンカー情報表記憶部にアンカー情報を登録するアンカー登録部をさらに備えたので、いつでも外部からアンカー情報の登録を行うことができることになり、利用状況に合わせたソフトウェア連携を実現でき、操作の向上が期待できる。

【0119】また、上記アンカー識別手段は、上記情報取得手段が取得した情報を蓄積する取得情報蓄積手段をさらに備え、上記アンカー照合部は、上記取得情報蓄積手段に蓄積された複数の情報の組合せによるアンカー照合を可能にするようにしたので、複数の情報の組み合わせをアンカー情報とすることにより、独自に設定した操作へのソフトウェア連携が可能になり、通常、行われな操作へ連携を登録することができ、操作誤りを原因とするソフトウェア連携の誤動作が減少する。

【0120】また、上記アンカー識別手段は、上記アンカー照合部で条件を満たすと判断された全てのアンカー情報を蓄積して上記連携実行手段に出力する連携アンカー蓄積部をさらに備えたので、複数のアンカー情報を蓄積することにより、複数の連携処理を実現できる。

【0121】また、上記連携実行手段は、ソフトウェア連携情報を登録して蓄積する連携情報表記憶部と、上記アンカー識別手段から得られたアンカー情報に対応するソフトウェア連携情報を上記連携情報表記憶部から取得する連携情報取得部と、上記連携情報取得部が取得したソフトウェア連携情報から連携先ソフトウェアに実行命令を出す連携情報実行部とを備えたので、連携元ソフトウェア自体に変更を加えることなくソフトウェア連携を実現できると共に、すべてのソフトウェアに対し個別にソフトウェア自体の変更なしに外部から連携処理を実現させることができる。

【0122】また、上記アンカー識別手段により識別または登録蓄積されたアンカー情報を画面に表示するアンカー表示手段をさらに備えたので、アンカー情報を画面表示することによりソフトウェア連携を行う場所を視覚的にユーザに伝えることができ、操作の向上が期待できる。

【0123】また、上記機能追加手段は、上記情報取得手段が取得した情報に基づいて対象ソフトウェアの動作が表示処理を行うトリガの場合に表示情報を出力する表示情報処理手段と、この表示情報処理手段からの表示情報を対象ソフトウェア上に表示する表示手段とを備えたので、対象ソフトウェアを監視し、対象ソフトウェアの動作が表示処理を行う際に表示情報を対象ソフトウェア上に表示することで、拡張機能のないソフトウェアに対して機能拡張を行うことができる。

【0124】また、操作情報を入力するための操作部と、操作命令情報を蓄積してなる操作命令情報記憶手段と、上記操作部から入力された操作情報に対応して実行すべき命令情報を上記操作命令情報記憶手段から検索し、検索された命令を実行する操作命令処理手段とをさらに備え、上記操作命令処理手段は、命令情報の中で対象ソフトウェア上に表示を行う際に、上記表示情報処理手段に表示処理を依頼するようにしたので、対象ソフトウェアの表示ウィンドウ上に表示される操作部を操作し、操作情報に対応して命令情報を検索し、検索された命令を実行することができる。

【0125】また、上記操作命令処理手段により検索された命令情報が対象ソフトウェアに対する命令である場合に、当該操作命令処理手段から出力される命令情報の入力に基づいて対象ソフトウェアへの命令を送信する対象ソフトウェア命令手段をさらに備えたので、対象ソフトウェアを操作している感覚で対象ソフトウェア自体の状態を変化させるような機能を対象ソフトウェア自体の変更を行うことなく追加させることができる。

【0126】また、対象ソフトウェア使用時に高い頻度で行う処理の手順情報を蓄積してなる処理手順情報記憶手段と、指定された処理手順を上記処理手順情報記憶手段から検索する処理手順検索手段とをさらに備え、上記操作命令処理手段は、上記操作部から入力された操作情報に対応する命令情報を検索し、操作命令情報が処理手順情報記憶手段に蓄積されている処理を行う命令であった場合に、上記処理手順検索手段に処理手順を指定して検索させるようにしたので、通常頻繁に実行を行う手順情報を保存しておき、拡張機能のないソフトウェアについてそのソフトウェア自体を操作する感覚でその手順情報を実行することができる。

【0127】また、操作命令情報記憶手段に蓄積する操作命令を追加する操作命令追加手段をさらに備えたので、利用者による新たな操作命令情報を追加し、また、その操作命令情報のトリガを発生させる操作部のオブジェクトを追加することができる。

【0128】さらに、他の発明に係る情報処理装置のソフトウェア連携システムは、複数のソフトウェアが動作可能なオペレーティングシステムを持つ情報処理装置であって、連携元ソフトウェアのファイルまたはウィンドウが表示する内容をアンカー識別できるソフトウェア形式に変換するファイル変換手段と、アンカー識別できるソフトウェアが識別したアンカー情報に対応して連携先ソフトウェアを実行する連携実行手段とを備えたので、アンカー識別できないソフトウェア形式のファイルやウィンドウ表示内容をアンカー識別できるソフトウェア形式のファイルに変換することにより、連携元ソフトウェア自体に変更を加えることなくソフトウェア連携を行うことができる。

【図面の簡単な説明】

【図1】 この発明に係る情報処理装置のソフトウェア連携システムを示す構成図である。

【図2】 マウス操作によるソフトウェア連携の画面例の説明図である。

【図3】 この発明の実施の形態1に係るソフトウェア連携システムの動作フローチャートである。

【図4】 オペレーティングシステムから情報取得を行う情報取得装置の内部構成図である。

【図5】 オペレーティングシステムから情報取得を行う情報取得装置の動作フローチャートである。

【図6】 アンカー識別装置の内部構成図である。

【図7】 アンカー識別装置の動作フローチャートである。

【図8】 ソフトウェア名をアンカー情報の種類とするアンカー情報表の説明図である。

【図9】 連携実行装置の内部構成図である。

【図10】 連携実行装置の動作フローチャートである。

【図11】 連携情報表の説明図である。

【図12】 透明ウィンドウが情報率得を行う情報取得装置の内部構成図である。

【図13】 透明ウィンドウが情報取得を行う情報取得装置の動作フローチャートである。

【図14】 予め設定した情報の取得を行う情報取得装置の内部構成図である。

【図15】 予め設定した情報の取得を行う情報取得装置の動作フローチャートである。

【図16】 ソフトウェアのオブジェクトの情報を取得する取得情報表の説明図である。

【図17】 ソフトウェアが外部に公開している変数から情報を受け取る取得情報表の説明図である。

【図18】 情報処理装置内のメモリの特定のアドレスの情報を取得する取得情報表の説明図である。

【図19】 ソフトウェアが作成するファイルから情報を取得する取得情報表の説明図である。

【図20】 イベントをアンカー情報の種類とするアンカー情報表の説明図である。

【図21】 座標をアンカー情報の種類とするアンカー情報表の説明図である。

【図22】 キー情報をアンカー情報の種類とするアンカー情報表の説明図である。

【図23】 ウィンドウ状態をアンカー情報の種類とするアンカー情報表の説明図である。

【図24】 オブジェクト名をアンカー情報の種類とするアンカー情報表の説明図である。

【図25】 オープン中のファイル名をアンカー情報の種類とするアンカー情報表の説明図である。

【図26】 ページ番号をアンカー情報の種類とするアンカー情報表の説明図である。

【図27】 図8、図20～26の全ての情報をアンカー情報の種類とするアンカー情報表の説明図である。

【図28】 情報送出部を備えるアンカー識別装置の内部構成図である。

【図29】 アンカー登録部を備えるアンカー識別装置の内部構成図である。

【図30】 取得情報蓄積部を備えるアンカー識別装置の内部構成図である。

【図31】 連携アンカー蓄積部を備えるアンカー識別装置の内部構成図である。

【図32】 アンカー表示装置を備えるソフトウェア連携システムの構成図である。

【図33】 他の発明に係る情報処理装置のソフトウェア連携システムを示す構成図である。

【図34】 図33に示すソフトウェア連携システムの動作フローチャートである。

【図35】 この発明の実施の形態2に係る情報処理装置のソフトウェア連携システムを示す構成図である。

【図36】 図35に示す情報取得手段200の内部構成図である。

【図37】 図35に示す情報取得手段200の動作フローチャートである。

【図38】 図35に示す表示情報処理手段201の内部構成図である。

【図39】 図35に示す表示情報処理手段201の動作フローチャートである。

【図40】 この発明の実施の形態22に係る情報処理装置のソフトウェア連携システムを示す構成図である。

【図41】 図40に示す操作部203の動作フローチャートである。

【図42】 図40に示す操作部203内の操作／操作ID変換テーブルの説明図である。

【図43】 図40に示す操作命令処理手段204の動作フローチャートである。

【図44】 図40に示す操作命令情報記憶手段205が有する操作ID毎の操作命令情報テーブルの説明図である。

【図45】 この発明の実施の形態23に係る情報処理装置のソフトウェア連携システムを示す構成図である。

【図46】 図45に示す操作命令処理手段204の動作フローチャートである。

【図47】 図45に示す対象ソフトウェア命令手段206の内部構成図である。

【図48】 図45に示す対象ソフトウェア命令手段206の動作フローチャートである。

【図49】 この発明の実施の形態24に係る情報処理装置のソフトウェア連携システムを示す構成図である。

【図50】 図49に示す操作部203内の操作／操作ID変換テーブルの説明図である。

【図51】 この発明の実施の形態24に係る動作フローチャートである。

【図52】 この発明の実施の形態25に係る情報処理装置のソフトウェア連携システムを示す構成図である。*

*【図53】 図52に示す操作命令追加手段209の内部構成図である。

【図54】 図52に示す操作命令追加手段209の動作フローチャートである。

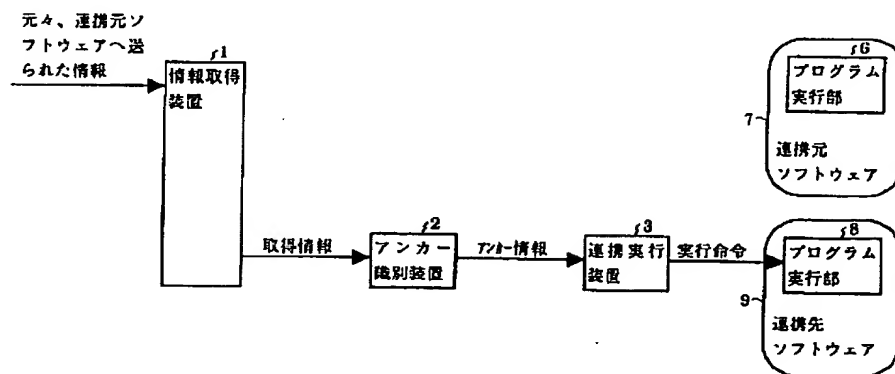
【図55】 図52に示す操作命令情報記憶手段205に備えられる各種テーブルの説明図である。

【図56】 従来例に係るOLEによるソフトウェア連携の画面例の説明図である。

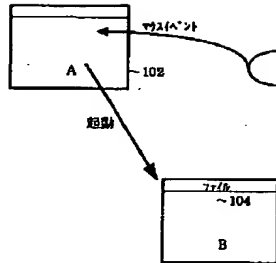
【符号の説明】

- 10 1 情報取得手段、2 アンカー識別装置、3 連携実行装置、4 アンカー表示装置、6、8 プログラム実行部、7 連携元ソフトウェア、9 連携先ソフトウェア、10 ファイル変換装置、19 情報取得部、20 取得情報表記憶部、21 情報取得登録部、22 情報受信部、26 透明ウィンドウ部、27 ウィンドウ情報取得部、31 アンカー照合部、32 アンカー情報表記憶部、33 情報送出部、34 アンカー登録部、35 取得情報蓄積部、36 連携アンカー蓄積部、41 連携情報取得部、42 連携情報実行部、43 連携情報表記憶部、200 情報取得手段、201 表示情報処理手段、202 表示手段、204 操作命令処理手段、205 操作命令情報記憶手段、206 対象ソフトウェア命令手段、207 処理手順検索手段、208 処理手順情報記憶手段、209 操作命令追加手段、210 情報取得登録部、211 情報受信部、212 情報解析部、213 表示情報作成部、214 表示手段制御部、215 参照テーブル、216 外部コマンドテーブル、217 内部コマンドテーブル、218 共通命令送信部、219 個別命令送信部、220 追加命令入力部、221 追加命令登録部、222 操作部追加部、223 対象ソフトウェア起動部、224、225 操作／操作ID変換テーブル。

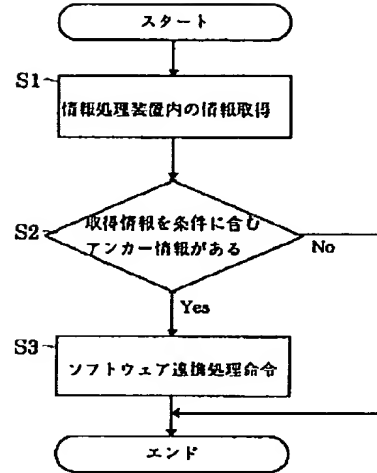
【図1】



【図2】



【図3】



【図8】

ソフトウェアID	ソフトウェア名
1	ソフトウェアA
2	ソフトウェアB

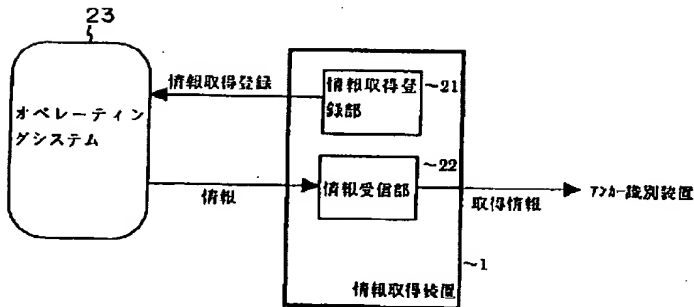
【図18】

取得アドレス	取得内容
0x40000000	ソフトウェアAの表示ページ番号
0x80000000	ソフトウェアBのオープンファイル名

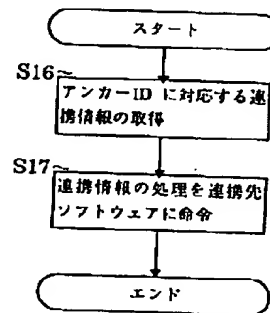
【図21】

ソフトウェアID	座標
1	x座標 ≤ 200 y座標 ≤ 50
2	500 ≤ x座標 ≤ 1000 200 ≤ y座標 ≤ 500

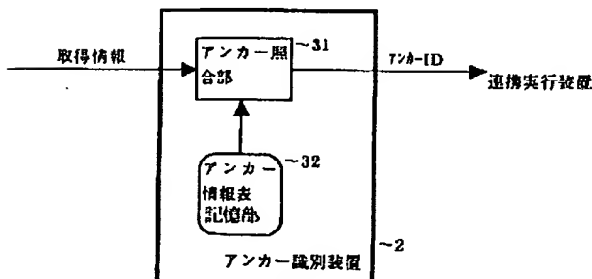
【図4】



【図10】



【図6】



【図11】

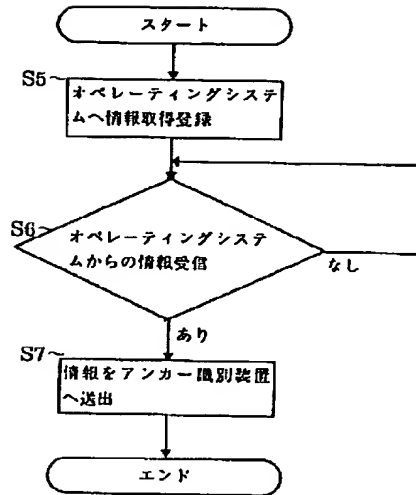
ソフトウェアID	連携先ソフトウェア名	オープンファイル名	ウィンドウ状態
1	ソフトウェアB	ファイルB1	
2	ソフトウェアC	ファイルC1	
3	ソフトウェアC	ファイルC2	最大化
4	ソフトウェアD		

空欄は情報が未設定であることを示す。

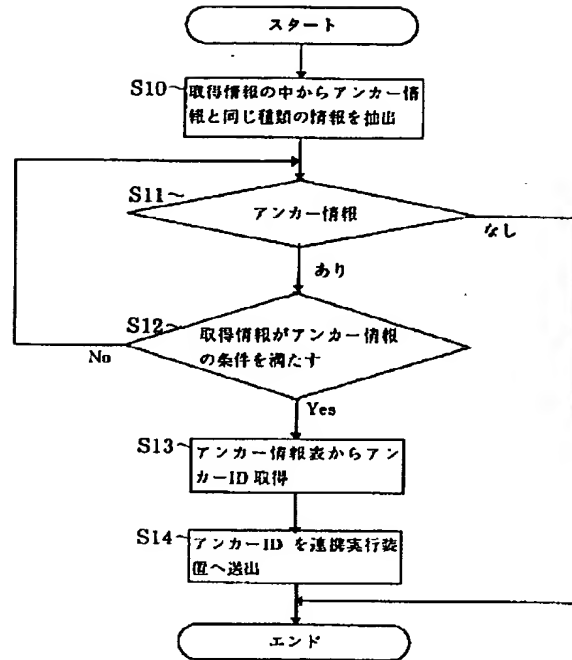
【図19】

ファイル名	取得内容
ProgramA.log	ソフトウェアAの表示ページ番号
ProgramB.tmp	ソフトウェアBのオープンファイル名

【図5】



【図7】



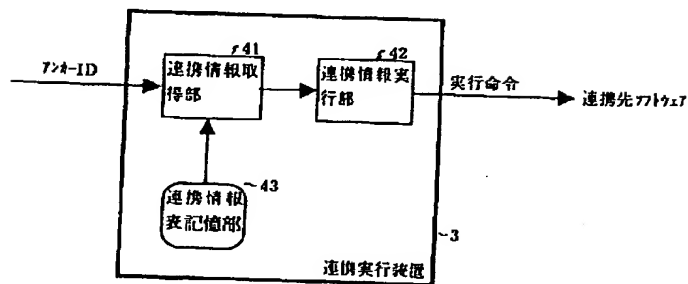
【図22】

【図23】

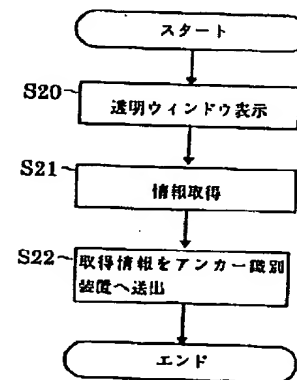
Fn-ID	キー情報
1	[Ctrl+A]
2	[Ctrl+B]

Fn-ID	ウィンドウ状態
1	アイコン化
2	最大化

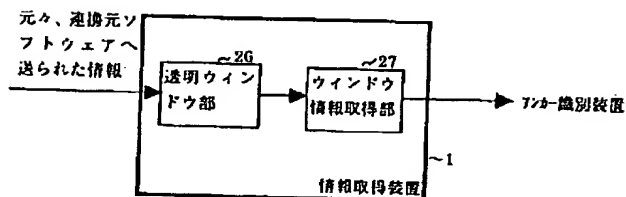
【図9】



【図13】



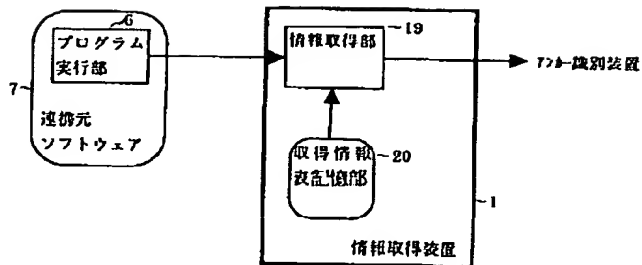
【図12】



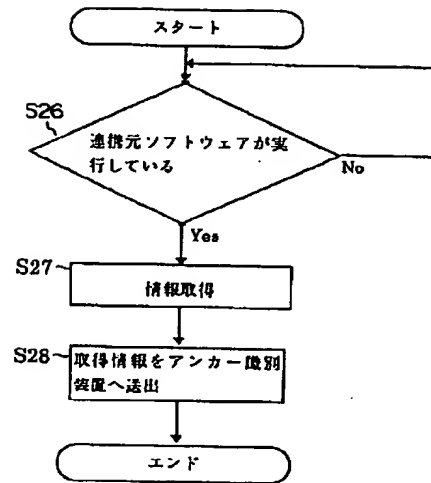
【図20】

Fn-ID	イベント
1	マウスボタンダウン
2	マウス移動
3	キー入力
4	ウィンドウアクティブ

【図14】



【図15】



【図16】

連携元ソフトウェア名	オブジェクト名	取得内容
ソフトウェアA	オブジェクトA1	表示ページ番号
ソフトウェアB	オブジェクトB1	オープンファイル名

【図17】

連携元ソフトウェア名	変数名	取得内容
ソフトウェアA	変数A1	表示ページ番号
ソフトウェアB	変数B1	オープンファイル名

【図24】

ファイルID	オブジェクト名
1	オブジェクトA1
2	オブジェクトB1

【図25】

ファイルID	オープンファイル名
1	ファイルA1
2	ファイルB1

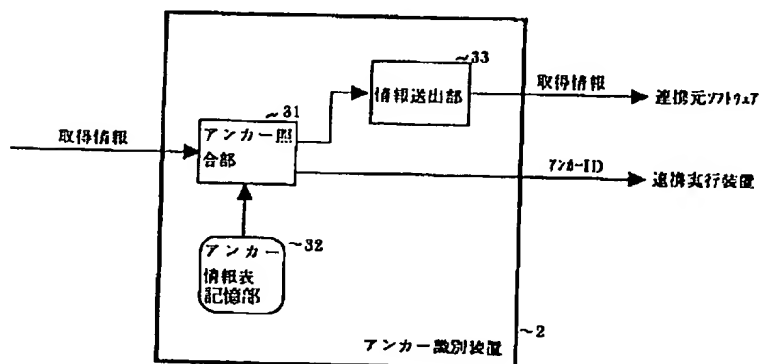
【図26】

ファイルID	ページ番号
1	1
2	3

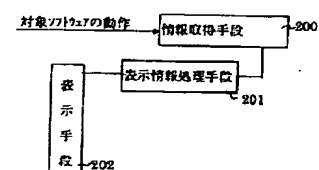
【図27】

ファイルID	ソフトウェア名	イベント	座標	ト-情報	アイコン状態	オブジェクト名	オープンファイル名	ページ番号
1	ソフトウェアA	マウスクリック	x座標 ≤ 200 y座標 ≤ 50					
2	ソフトウェアB				アイコン化			
3	ソフトウェアB					オブジェクトB1	ファイルB1	2
4		キー入力		[Ctrl+9]				

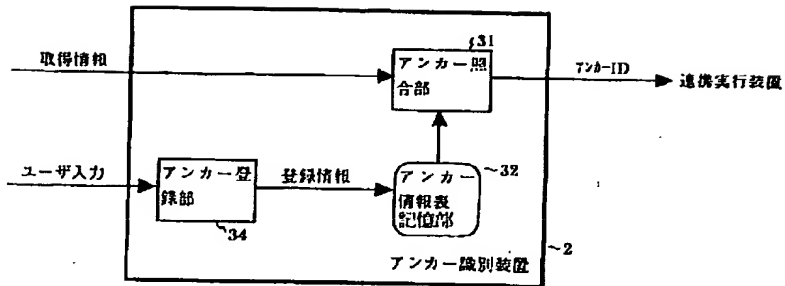
【図28】



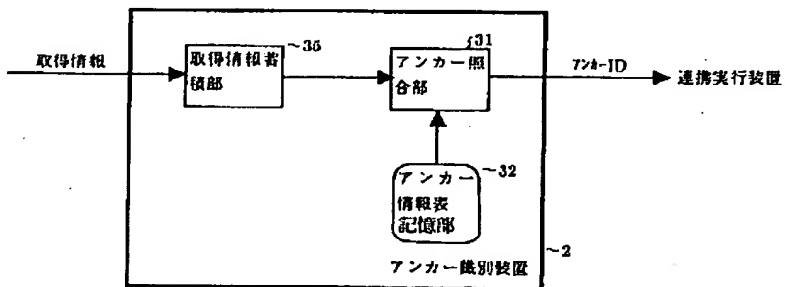
【図35】



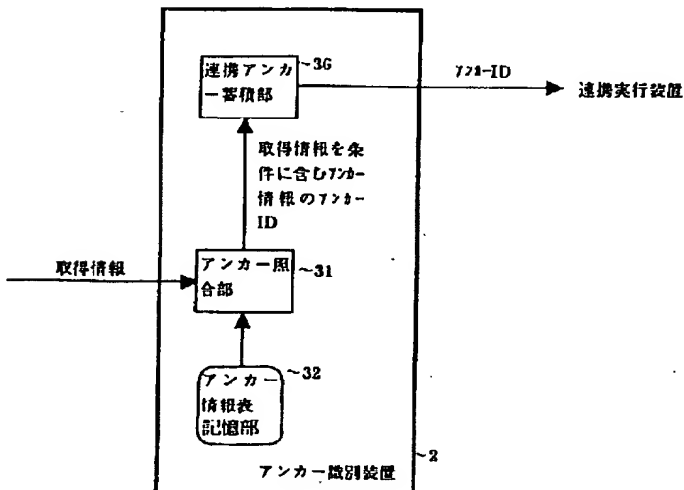
【図29】



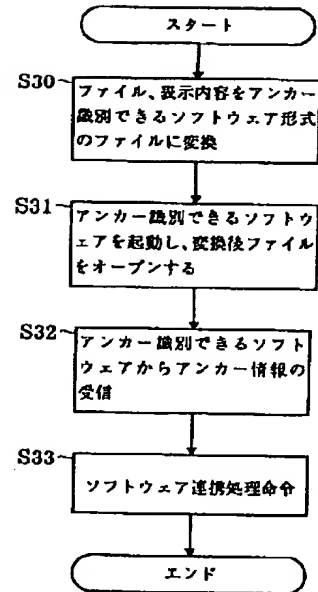
【図30】



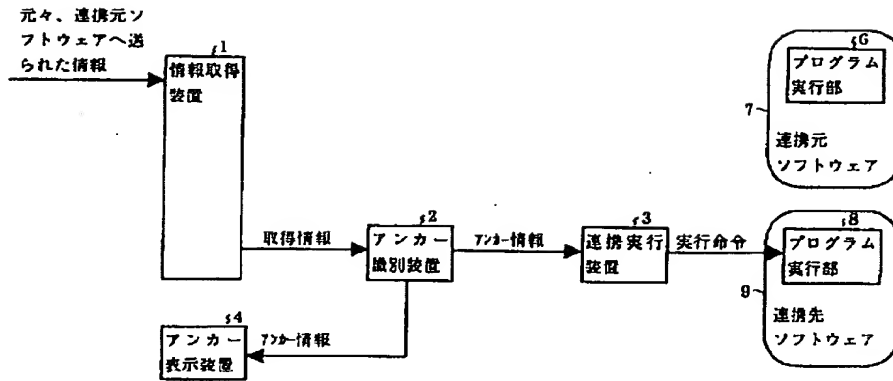
【図31】



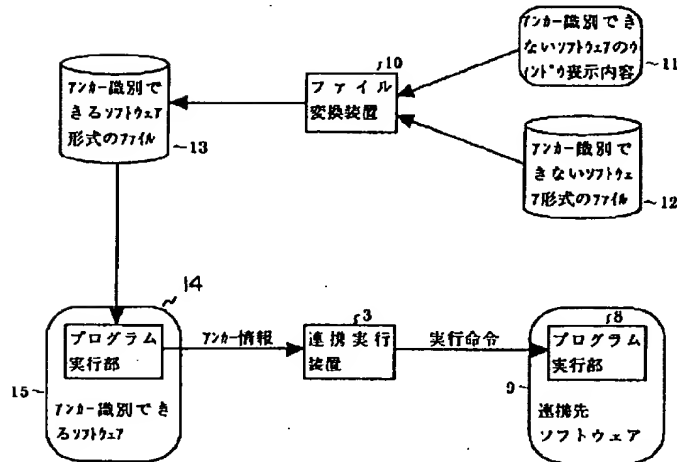
【図34】



【図 32】



【図 33】

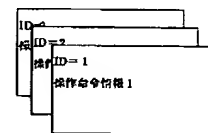


【図 42】

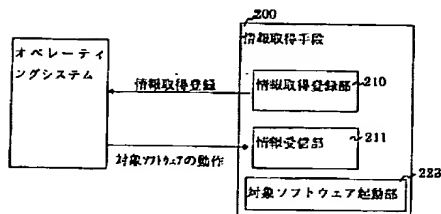
操作/操作ID変換テーブル 224

操作オブジェクト	イベント	操作ID
ボタン1	Click	ID1
テキスト1	Click	ID2
テキスト1	Edit	ID3
:	:	:
:	:	:

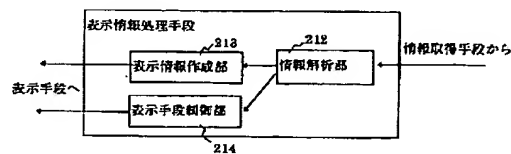
【図 44】



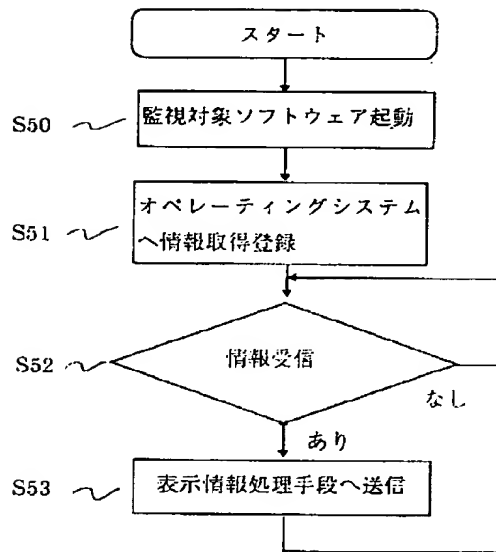
【図 36】



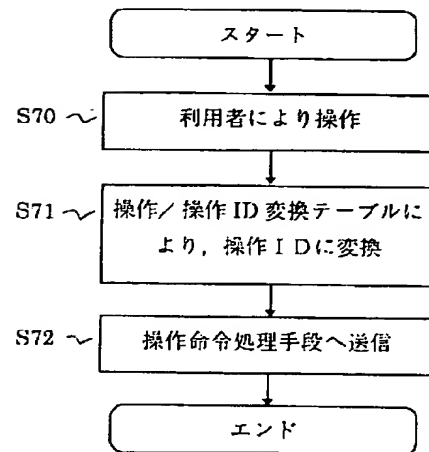
【図 38】



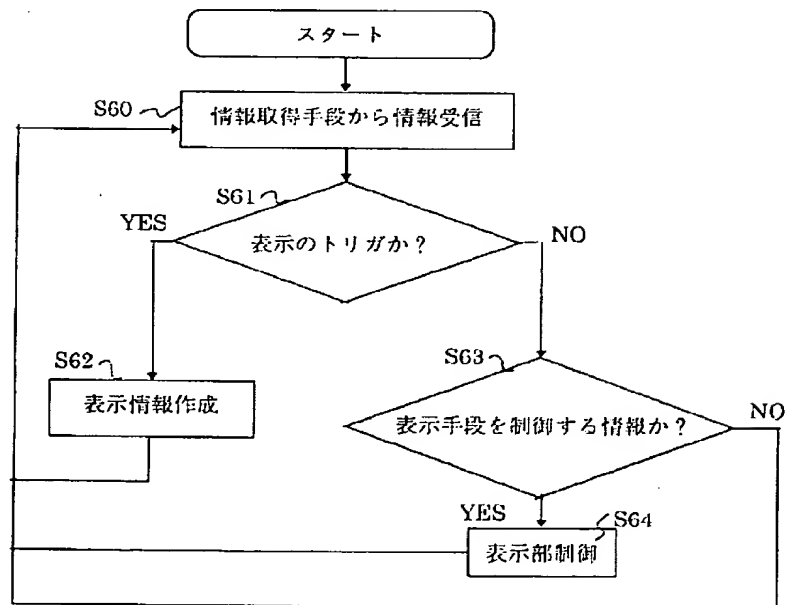
【図37】



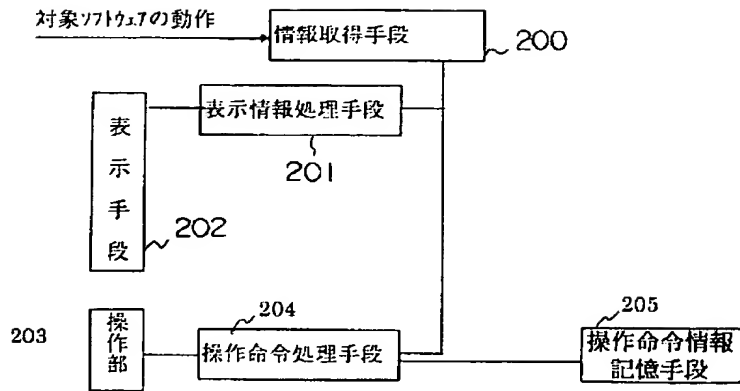
【図41】



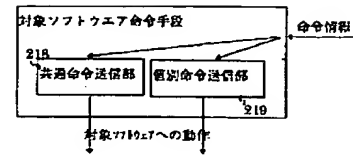
【図39】



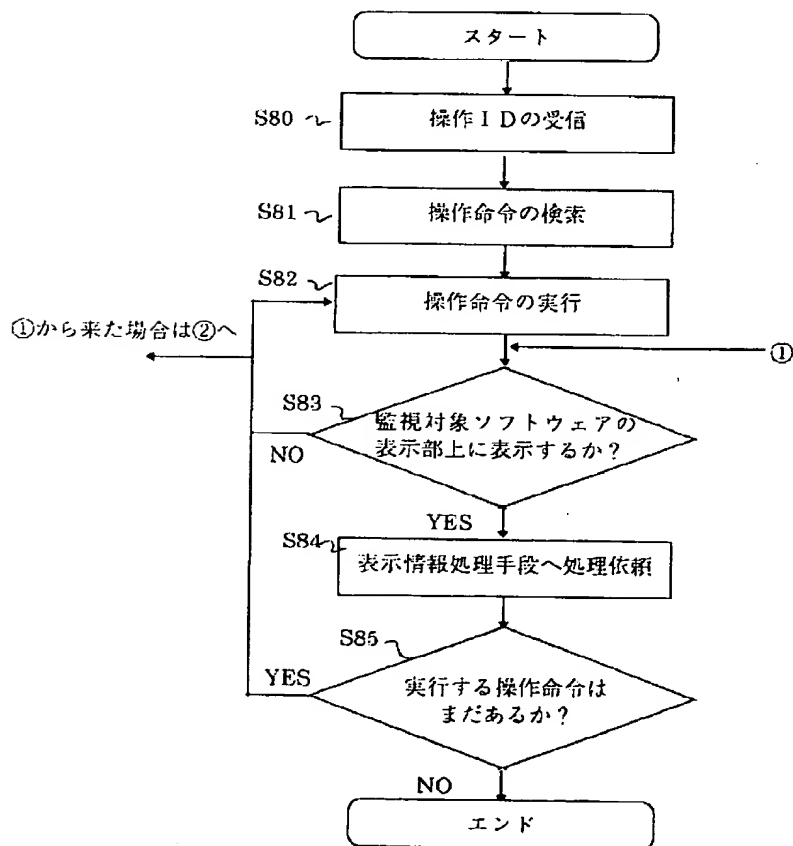
【図40】



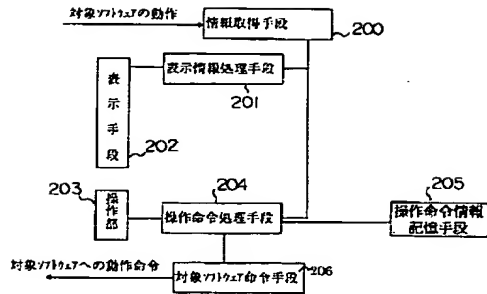
【図47】



【図43】



【図45】

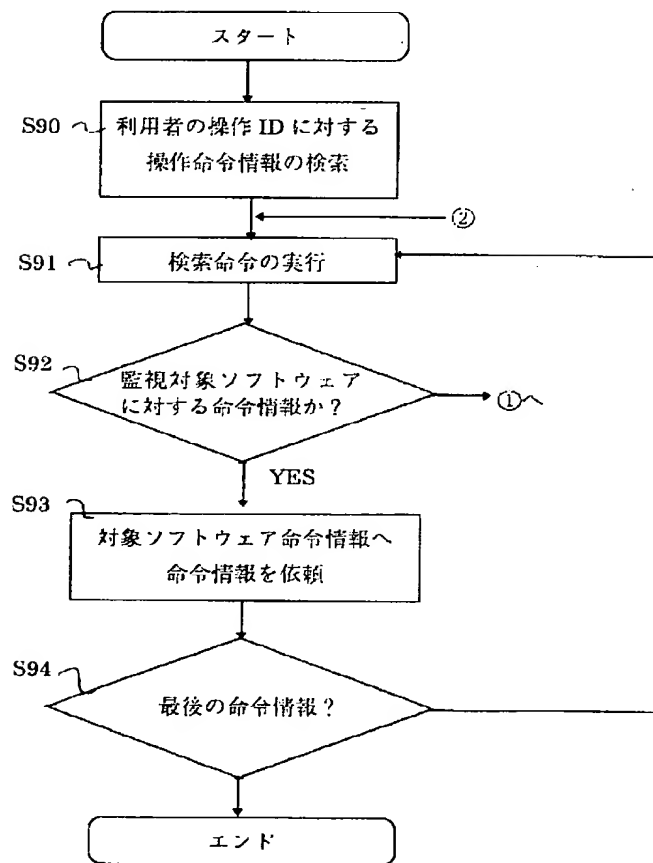


【図50】

操作/操作ID変換テーブル 225

操作オブジェクト	イベント	操作ID	手順
ボタン1	Click	ID1	NO
テキスト1	Click	ID2	YES
テキスト1	Edit	ID3	NO
:	:	:	:
:	:	:	:

【図46】




```
graph TD; Start([スタート]) --> S100[S100 命令の受信]; S100 --> S101{S101 共通命令情報か?}; S101 -- YES --> S102[S102 共通命令送信部へ処理依頼]; S101 -- NO --> S103[S103 個別命令送信部へ処理依頼]; S102 --> End([エンド]); S103 --> End;
```

FIG. 1 is a flowchart illustrating the command reception process. The process begins with a start block (スタート), followed by a process block (S100) labeled "命令の受信" (Command Reception). This leads to a decision block (S101) labeled "共通命令情報か?" (Is it common command information?). If the answer is YES, the process flows to a process block (S102) labeled "共通命令送信部へ処理依頼" (Request processing to common command transmission section). If the answer is NO, the process flows to a process block (S103) labeled "個別命令送信部へ処理依頼" (Request processing to individual command transmission section). Both S102 and S103 lead to an end block (エンド).

```

graph TD
    A[対象ソフトウェアの動作] --> B[情報取得手段 200]
    B --> C[表示情報処理手段 201]
    C --> D[表示手段 202]
    E[操作部 203] --> F[操作命令処理手段 204]
    F --> B
    F --> G[操作命令情報記憶手段 205]
    G --> H[処理手段検索手段 207]
    H --> I[処理手順情報記憶手段 208]
    I --> J[対象ソフトウェアへの動作命令 206]
    J --> K[対象ソフトウェア命令手段 209]
    K --> F

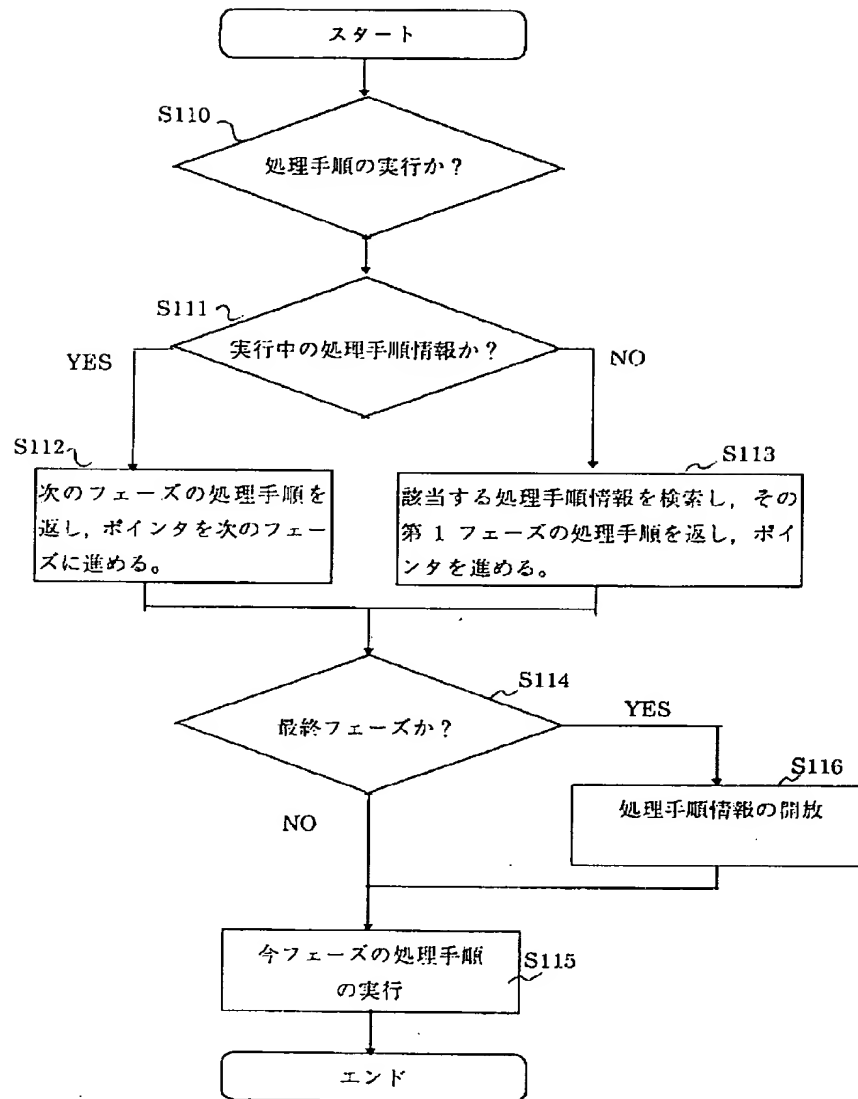
```

```

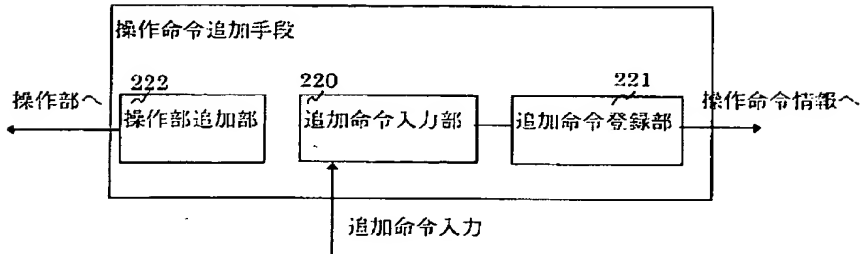
graph TD
    200[情報取得手段] --> 201[表示情報処理手段]
    201 --> 202[表示手段]
    202 --> 203[操作部]
    203 --> 204[操作命令処理手段]
    203 --> 209[操作命令追加手段]
    204 --> 205[操作命令情報記憶手段]
    205 --> 200
  
```

対象771927の動作

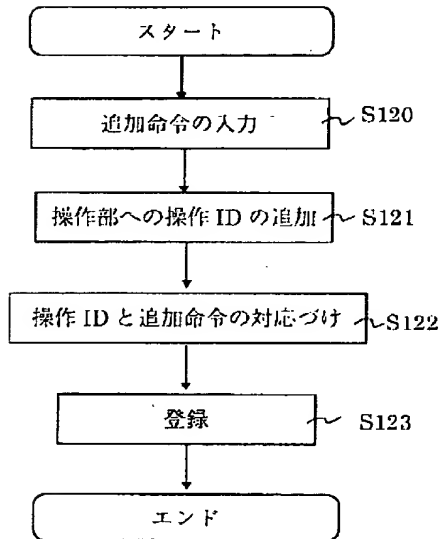
【図51】



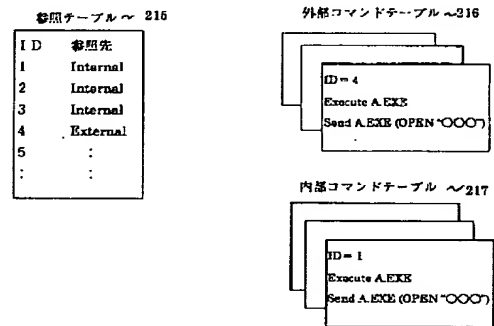
【図53】



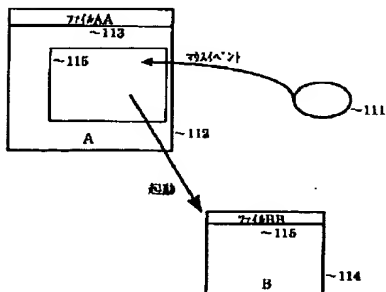
【図54】



【図55】



【図56】



(27)

特開平 1 1 - 2 4 2 6 0 0

フロントページの続き

(72)発明者 横里 純一
東京都千代田区丸の内二丁目 2 番 3 号 三
菱電機株式会社内

(72)発明者 鈴木 靖宏
東京都千代田区丸の内二丁目 2 番 3 号 三
菱電機株式会社内